

**SELF-CALIBRATING PARTICIPATORY
WIRELESS INDOOR LOCALIZATION**

CHENGWEN LUO

NATIONAL UNIVERSITY OF SINGAPORE

2015

**SELF-CALIBRATING PARTICIPATORY
WIRELESS INDOOR LOCALIZATION**

CHENGWEN LUO

B.Eng.

A DISSERTATION SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

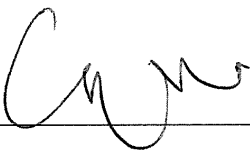
DEPARTMENT OF COMPUTER SCIENCE
SCHOOL OF COMPUTING
NATIONAL UNIVERSITY OF SINGAPORE

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

A handwritten signature in black ink, appearing to be 'Cw Luo', is written over a horizontal line.

Chengwen Luo

September 15, 2015

Acknowledgment

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Mun Choon Chan, for his guidance and support throughout my Ph.D study in NUS. He is always very nice and will always be there for his students. I still remember that before many paper deadlines, we worked late together improving our papers. Those are valuable memories that I will never lose. He keeps inspiring me with his profound insights and immense knowledge. The work would not have been possible without him. I could not have imagined having a better advisor for my Ph.D study.

I am grateful to my dear lab mates, Shao Tao, Xiangfa, Naba, Manjunath, Hwee Xian, Fai Cheong, Hande, Kartik, Mobashir, Girisha, Chaodong, Yu Da, Liu Xiao, Nimantha, Wang Hui, Pravein, and others. I have been greatly inspired by them, and they made our lab a joyful place to be during study, and a place to miss after leaving.

Many thanks to Prof. Ananda, for sharing his thoughts about research and his philosophy on life, and for bringing candies to the lab to brighten our days. Thanks to Prof. Seth Gilbert, Prof. Ooi Wei Tsang, and Prof. Ben Leong, who gave valuable feedback on my research. I would like to thank the anonymous reviewers of all the conferences and journals we submitted to, for all their insightful comments. I am also thankful to all the participants in our experiments, for making those experiments possible.

I would like to express my sincere thanks to my dear friends: Ye Nan, who has given me so much help with my research and my life, and Zhiqiang, Jianxing, Zhuolun, Kegui, and Zhai Jing, for all the food, play, sharing and support, and friends who made my life much more colorful during my graduate studies: Zhang Li, Weiwei, Gan Tian, Liu Shuang, Chen Tao, Fang Da, Wendy, Siqu, Pei Ying, Cheng Long, Chen Ju, and many others.

Without the support of my family it would never have been possible for me to finish my Ph.D studies. The selfless love of my parents, my brother, and my grandparents has made me who I am today. No words can express my love for them, and so I dedicate this thesis to them.

Finally, I want to thank Wenjun. I have always felt blessed to have met a wonderful person like her, and thank her for her support during my Ph.D studies and the happiness she brings to my life.

September 15, 2015

Contents

Contents	v
List of Tables	xi
List of Figures	xiii
1 Introduction	1
1.1 Wireless Indoor Localization	1
1.2 Participatory Sensing Based Indoor Localization	2
1.3 Overview of the Proposed Approaches	3
1.3.1 PiLoc: Self-calibrating Active Indoor Localization	3
1.3.2 SpiLoc: Self-calibrating Passive Indoor Localization	4
1.3.3 A^2Loc : Accuracy Awareness of Wireless Indoor Localization	5
1.4 Contributions	7
1.5 Thesis Structure	7
2 Literature Review	9
2.1 Active Indoor Localization	9
2.1.1 Infrastructure Based Localization	9
2.1.2 Fingerprint Based Localization	10
2.1.3 Propagation Model Based Localization	10
2.1.4 SLAM Based Localization	11
2.1.5 Participatory Sensing Based Localization	11
2.2 Passive Indoor Localization	12
2.2.1 Device-free Passive Localization	12
2.2.2 Device-based Passive Localization	13
2.3 Wireless Signal Modeling	14

3	PiLoc: Self-calibrating Active Indoor Localization	15
3.1	Introduction	15
3.2	PiLoc Active Indoor Localization System	16
3.2.1	Overview of PiLoc	16
3.2.2	Data Collection	18
3.2.2.1	Fingerprint Collection	18
3.2.2.2	Inertial Sensing	18
3.2.3	Trajectory Clustering	19
3.2.3.1	AP Clustering	19
3.2.3.2	Floor Clustering	20
3.2.3.3	Path Segment Clustering	27
3.2.4	Trajectory Matching	28
3.2.4.1	Path Correlation	28
3.2.4.2	Signal Correlation	30
3.2.4.3	Final Matching	31
3.2.5	Floor Plan Construction	32
3.2.5.1	Algorithm	32
3.2.5.2	Floor Plan Filtering	35
3.2.5.3	Floor Plan Evolution	36
3.2.5.4	PiLoc Localization	36
3.2.6	Energy Management	37
3.2.6.1	WiFi Scanning Modes	37
3.2.6.2	Sensor-triggered WiFi Scanning	38
3.3	Performance Evaluation of PiLoc	41
3.3.1	Implementation	41
3.3.2	Data	41
3.3.3	Performance	42
3.3.3.1	Evaluation Metrics	42
3.3.3.2	Trajectory Clustering	43
3.3.3.3	Floor Plan Construction	44
3.3.3.4	Localization	45
3.3.3.5	Power consumption	46
3.4	Discussions	47

3.4.1	Applications	47
3.4.2	Limitations	48
3.4.3	Extensions	48
3.4.3.1	Diverse Floor Plans	48
3.4.3.2	Enriching Constructed Floor Plans	48
3.4.3.3	Multiple Fingerprints	49
3.5	Summary	49
4	SpiLoc: Self-calibrating Passive Indoor Localization	51
4.1	Introduction	51
4.2	SpiLoc Passive Indoor Localization System	53
4.2.1	Overview	53
4.2.1.1	System Architecture	53
4.2.1.2	Opportunistic Data Collection	55
4.2.2	Passive Landmarks	55
4.2.2.1	Passive Landmarks: Concept	55
4.2.2.2	Passive Landmarks: Identification	57
4.2.3	Trace Mapping	58
4.2.3.1	Walking Route Inference	58
4.2.3.2	Fingerprint Database Bootstrapping	61
4.2.3.3	Noise Filtering	62
4.2.3.4	SpiLoc Localization	64
4.3	Performance Evaluation of SpiLoc	65
4.3.1	System Implementation	65
4.3.2	Evaluation	65
4.3.2.1	Experiment Design	65
4.3.2.2	RSS Trace Mapping Performance	66
4.3.2.3	Impact of Sparsity of Transmission Detections	69
4.3.2.4	Impact of Variations in the Walking Speed	70
4.3.2.5	Localization Performance	71
4.4	Discussion	73
4.4.1	Dedicated Site Surveys	73
4.4.2	Prompting Extra Transmissions	73
4.4.3	Open Area	74

4.4.4	Privacy Risks	74
4.5	Summary	74
5	A^2Loc: Accuracy Awareness of Wireless Indoor Localization	77
5.1	Introduction	77
5.2	Accuracy Awareness	79
5.2.1	Preliminaries	79
5.2.2	Accuracy Awareness	82
5.2.2.1	Point-level Accuracy	83
5.2.2.2	Region-level Accuracy	87
5.2.2.3	Floor-level Accuracy	89
5.3	Performance Evaluation of A^2Loc	92
5.3.1	Data	92
5.3.2	Performance	93
5.3.2.1	Error Estimation	93
5.3.2.2	Landmark Detection	93
5.3.2.3	BSSID Subset Selection	94
5.3.2.4	Localization Algorithm Selection	95
5.4	Summary	95
6	Conclusion and Future Work	97
6.1	Research Contributions	98
6.1.1	PiLoc: Self-calibrating Active Indoor Localization	98
6.1.2	SpiLoc: Self-calibrating Passive Indoor Localization	98
6.1.3	A^2Loc : Accuracy Awareness of Fingerprint-based Wireless Indoor Localization	99
6.2	Future Work	99
	Bibliography	103

Summary

Knowing the accurate indoor location is often critically important to many mobile applications. However, despite significant progress, an indoor localization system that can be easily deployed on a large scale remains a challenge. One important obstacle hindering the large-scale deployment of existing indoor localization systems is labor-intensive site survey and system maintenance. Many of these systems involve a dedicated offline calibration stage that builds a radio map to aid localization. In addition, they also need to be periodically updated to reflect environmental changes. Another challenge is the lack of systematic performance evaluation approaches. As a result, it is hard to deploy and maintain fingerprint-based wireless indoor localization systems in practice.

In view of these deployment and evaluation challenges, the focus of the work described in this thesis is to effectively tackle these challenges by designing accuracy-aware self-calibrating localization systems. There are three major contributions in this thesis: (1) We design and implement *PiLoc*, a self-calibrating active indoor localization system, which infers the indoor maps and outputs radio maps for localization automatically through merging participatory sensing input. (2) To enable localization without the explicit cooperation of mobile devices, we design and implement *SpiLoc*, which focuses on passive localization for mobile devices. *SpiLoc* automatically bootstraps the passive fingerprint database for localization through opportunistic received signal strength (RSS) trace mapping. (3) We propose *A²Loc*, which introduces *accuracy awareness* to fingerprint-based indoor localization systems. *A²Loc* takes the radio maps generated from fingerprint-based indoor localization systems as input and outputs the estimated accuracy levels for these systems. These three systems are summarized below:

PiLoc. Unlike other current state-of-the-art systems, *PiLoc* leverages participatory sensing to bootstrap the active localization database while requiring no prior knowledge of the indoor environment. The key novelty of *PiLoc* is that it merges the crowdsourcing input annotated with sensor readings and WiFi signal strengths to generate the map of the indoor environment and construct the fingerprint database automatically. This self-calibrating capability makes

PiLoc practical and much easier to deploy and maintain without requiring prior knowledge of the indoor environment and dedicated site-surveys. The evaluation shows that PiLoc is able to work in various types of indoor environments and can achieve localization accuracy comparable to that of systems that require dedicated calibration, with 80% localization error less than 3 meters.

SpiLoc. SpiLoc is a passive indoor localization system that requires no collaboration from mobile devices. The key novelty of SpiLoc is that it leverages the novel RSS trace mapping technique to dynamically map the captured RSS traces to indoor pathways. The mapping automatically bootstraps the passive fingerprint database for localization. To the best of our knowledge, SpiLoc is the first participatory sensing based passive localization system to have the self-calibrating capability and provide fine-grained passive localization.

A²Loc. *A²Loc* exploits a Gaussian process based approach that uses as input the radio map collected and localization algorithm to be evaluated, and outputs the expected accuracy of the system. In addition, *A²Loc* provides useful information such as localization landmarks that can be used to further improve the localization accuracy. To the best of our knowledge, *A²Loc* is the first to achieve accuracy awareness in fingerprint-based localization systems. With this capability, it has the potential to be integrated into future fingerprint-based localization systems as a standard component to provide direct feedback about the accuracy level and guidelines in order to achieve better accuracy.

Overall, for this thesis, we designed and implemented a systematic solution for self-calibrating indoor localization systems. Of the proposed solutions, PiLoc and SpiLoc provide fine-grained localization for both active and passive localization, and *A²Loc* further improves the practicability by providing direct accuracy estimations. The proposed systems advance the current state-of-the-art systems by incorporating participatory sensing to provide accuracy-aware self-calibrating indoor localization systems, which significantly reduce calibration and maintenance costs and have the potential for large-scale deployment.

Keywords: Indoor Localization, Self-calibrating, Participatory Sensing, Accuracy Awareness

List of Tables

2.1	State-of-the-art Indoor Localization Systems	14
3.1	Performance of Barometer-based Floor-Transition Detection When Using Stairs	43
3.2	Performance of Barometer-based Floor-Transition Detection When Using Elevators	43
3.3	Floor Clustering Performance	44
3.4	Listing of related localization systems	46
3.5	Power Consumption Measurement	47
4.1	Landscape of Indoor Localization Research	52
4.2	Comparison with Different Localization Schemes	72

List of Figures

1.1	Overview of the works proposed in this thesis	6
3.1	Overview of PiLoc	17
3.2	Examples of Trajectories and Clustering	20
3.3	AP Clustering	20
3.4	Altitude behaviors during different floor transition events. Floor transition separates trajectory τ into different floor segments . .	22
3.5	Floor Constrain Update	24
3.6	Path Segment Clustering	27
3.7	CDF of Path Correlation	28
3.8	CDF of Signal Correlation	29
3.9	Stability of Signal Trends (Phone Varying)	30
3.10	Stability of Signal Trends (Time Varying)	30
3.11	ROC Curve of Final Matching	32
3.12	Example of Motion Vector Merging. d_{ij} denotes the current dis- placement and d'_{ij} denotes the new displacement.	34
3.13	Intra Trajectory Merging	35
3.14	Inter Trajectory Merging	35
3.15	Floor Plan Evolution	36
3.16	Floor Plan Construction for Various Indoor Environments	36
3.17	WiFi Signal Graph	38
3.18	Sensor-triggered WiFi Scanning	39
3.19	Heading Noise Detection	40
3.20	Multi-floor Floor Plan Construction	44
3.21	CDF of SME ($900m^2$ Office Floor)	45
3.22	CDF of SME ($120 m^2$ Research Lab)	45

3.23	Power Profile of PiLoc in Different States	47
3.24	CDF of LLE	47
4.1	System Architecture	54
4.2	Passive RSS Trend	56
4.3	Passive Landmarks	57
4.4	Different RSS Peaks When Walking Indoors	58
4.5	Route Generation Between Two Landmarks	58
4.6	RSS Evolution Pattern Comparison	60
4.7	RSS Divergence Change with Walking-Speed Variation	63
4.8	WiFi Monitor	65
4.9	Layout of the Testbed	66
4.10	Trace Mapping and Fingerprint Database Bootstrapping	67
4.11	Trace Mapping Performance For Traces Without Walking Speed Variation	68
4.12	CDF of Mapping Accuracy For All Landmark Pairs	68
4.13	Impact of Sparsity of Detection	69
4.14	Trace Mapping For Traces with Speed Variations	70
4.15	Performance of Variation Filtering	70
4.16	Localization Performance	71
4.17	Localization Error with Different Input Data	72
5.1	Mean Prediction ($\mu_{\mathbf{x}^* \mathcal{D}}$)	81
5.2	Variance Prediction ($\sigma_{\mathbf{x}^* \mathcal{D}}^2$)	81
5.3	Gaussian Process Sampling	82
5.4	Ground Truth Phone Sampling	82
5.5	Region Error Evolution	86
5.6	BSSID Selection ($240m^2$ Open Area)	91
5.7	BSSID Selection ($72m^2$ Office Room)	91
5.8	CDF of Point-level Error	92
5.9	Landmark Detection	94
5.10	Localization Algorithm Selection	94

Chapter 1

Introduction

1.1 Wireless Indoor Localization

Location is one of the most important types of context information in mobile and ubiquitous computing. Recently, wireless indoor localization has been the subject of extensive research efforts [86, 76, 63, 10, 78, 79, 81, 50, 19, 11, 74] due to both the need to support indoor location-based services, and the fact that GPS does not work well indoors. However, despite significant progress, developing an indoor localization system that can be easily deployed on a large scale remains a challenge.

One important obstacle that hinders the large-scale deployment of existing indoor localization systems is labor-intensive site survey and system maintenance. Many of these systems involve a dedicated offline calibration stage that builds a radio map to aid localization. This calibration stage involves the manual association of a location to be localized with its corresponding radio fingerprints. Furthermore, this radio map needs to be periodically updated to reflect changes in the environment. The calibration and maintenance effort required makes these systems tedious and difficult to deploy on a large scale.

Another challenge is the lack of systematic evaluation approaches. The settings of each existing indoor localization system are evaluated with different physical layouts and environmental effects, making it difficult to understand their performance and compare different localization systems directly. In particular, in localization systems where training data is mainly collected through crowdsourcing, an efficient evaluation approach is required to provide immediate

feedback regarding the accuracy levels.

Facing the challenges and deployment and evaluation, the focus of this thesis is effectively tackling these issues that affect the practicality of wireless indoor localization systems. We show that the calibration effort can be significantly reduced for both active localization and passive localization systems by exploiting participatory sensing. By merging the crowdsourcing sensing data, the systems are able to achieve self-calibrating capability to bootstrap themselves without dedicated site-surveys. In addition, by modeling the signal strength distribution using the constructed radio maps, the expected localization error of each indoor location can be obtained directly, hence achieving *accuracy awareness* and enabling systematic evaluation for wireless indoor localization systems.

1.2 Participatory Sensing Based Indoor Localization

Recently, *participatory sensing* [17] has been proposed as a new computing paradigm in mobile computing, and has been the subject of many research efforts [18, 41, 42, 43, 55, 64]. The idea of participatory sensing is to exploit the everyday mobile devices, such as smartphones, to form an interactive and collaborative sensing network that enables users to gather, share and analyze local knowledge [17]. By assigning sensing tasks to the ‘grassroots’ mobile devices, large-scale sensing systems and complex sensing applications can be enabled, covering different areas such as environment monitoring [55, 64], transportation [37], social networking[51], health care[44], etc.

Recognizing the effectiveness of participatory sensing, researchers have recently started to implement this idea in wireless indoor localization. Participatory sensing is used both to improve the localization accuracy [76, 32] and to reduce the calibration effort [63, 86, 74]. To improve the localization accuracy, crowdsourcing sensor data are merged to infer landmarks that are present in the indoor environment, to reduce localization errors [76]. With more users participating in this localization process, events involving social contacts such as encounter events can also be leveraged to reset the localization errors, improving the localization accuracy [32]. On the other hand, as more smartphone users participate in the data collection process, the input data can be used to construct the radiomaps that are required for localization, assuming accurate

floor plans and reliable landmarks are available [63, 86, 74]. Such approaches are able to efficiently reduce the calibration effort required, therefore making indoor localization systems more scalable and deployable.

However, accurate floor plans and sufficient numbers of reliable landmarks are not always easily available to reduce the calibration effort, and this assumption is one of the limitations of existing participatory sensing based indoor localization systems. In this thesis, we focus on the localization techniques that can significantly reduce the calibration effort to achieve self-calibration capability, while minimizing the assumption on the knowledge of the indoor environment. In addition, to assess the performance of a participatory sensing based indoor localization system, we also propose a systematic evaluation method to provide immediate feedback on the accuracy levels, based on current collected input data from participating users.

1.3 Overview of the Proposed Approaches

The following sections provide an overview of the three proposed systems and approaches, *PiLoc*, *SpiLoc*, and *A²Loc*, which were designed and implemented for this work.

1.3.1 PiLoc: Self-calibrating Active Indoor Localization

In *active indoor localization*, devices actively participate in the localization process to provide information obtained locally in order to infer the current indoor location. Existing active indoor localization systems [12, 88, 26, 20, 47] mostly rely on the uniqueness of WiFi signal strengths at different indoor locations, which is also known as WiFi fingerprinting [12], to determine the location of mobile devices. Compared with infrastructure-based localization schemes [62, 81], WiFi fingerprint-based indoor localization leverages existing infrastructures and is cheap and cost-effective, which makes it promising for large scale deployment. However, as many of these systems involve a dedicated offline calibration stage to build radio maps for the indoor environment, the deployment becomes time consuming and labor-intensive. To address this problem, participatory sensing based indoor localization systems [63, 86, 76, 74] have been proposed to exploit crowdsourcing to reduce the calibration overhead. Despite significant reduction

in the calibration effort and deployment effort, such systems rely heavily on the knowledge of the indoor floor, such as that provided by accurate floor plans [63, 86] and localization landmarks [76, 74], which is usually not easily available in practice.

On the other hand, PiLoc utilizes opportunistically sensed data contributed by participating users, while requiring no manual calibration, prior knowledge, or infrastructure support. The key novelty of PiLoc is that it merges automatically generated walking trajectories annotated with displacement and signal strength information from users to derive a map of walking paths annotated with radio signal strengths. With the generated indoor maps annotated with signal information, radio maps for localization are built automatically. Unlike previous systems, PiLoc does not require any knowledge of the indoor environment and maintains itself automatically, hence achieving self-calibrating capability. As PiLoc requires minimal user effort to calibrate and maintain, it has potential for large-scale deployment.

We implemented PiLoc and evaluated the system over five different indoor areas covering 5800 m^2 in total. The sizes of these five different floors ranged from 120 m^2 to 3000 m^2 . The smallest area of 120 m^2 was the inside of a research lab with lots of partitions, which posed a special challenge due to its very short turns and walk-ways. The evaluation shows that PiLoc was able to work in different types of indoor environments, and could achieve localization accuracy that comparable to that of systems that require dedicated calibration, with 80% localization error less than three meters.

1.3.2 SpiLoc: Self-calibrating Passive Indoor Localization

Passive indoor localization for smartphones enables a new spectrum of applications such as user tracking, mobility monitoring, social pattern analysis, etc. Unlike active localization, passive localization does not require the explicit participation of humans or devices, and usually relies on the opportunistic overhearing of packets transmitted by smartphones [56]. Since WiFi-enabled devices transmit wireless packets either intentionally for communication or unconsciously from background services, smartphones become trackable using WiFi monitoring devices without being connected to any specific WiFi APs or having any mobile

apps installed. Several passive localization systems have recently been proposed [56, 82, 83]. However, despite the fact that these existing systems have illustrated the feasibility of tracking multiple mobile devices passively, they either achieve coarse-grained localization accuracy with a localization error of about 70 meters[56], or require expensive infrastructure support [82, 83].

We therefore propose SpiLoc, a self-bootstrapped system for fine-grained passive indoor localization using non-intrusive WiFi monitors. SpiLoc uses off-the-shelf access point hardware to opportunistically capture WiFi packets to infer the location of smartphones in an indoor environment. The key novelty of SpiLoc lies in the fact that the passive fingerprint database for localization is automatically constructed and updated without any active participation of WiFi devices or manual calibration. To achieve this, SpiLoc first identifies passive landmarks that are present in WiFi received signal strength (RSS) traces. Given knowledge of the indoor floor plan and the location of WiFi monitors, SpiLoc statistically maps the collected RSS traces to specific indoor pathways. With sufficient mapping opportunistically detected, SpiLoc is able to automatically bootstrap a fine-grained passive fingerprint database for localization without requiring any additional calibration effort.

By mapping the RSS traces collected between different passive landmarks, SpiLoc bootstraps the passive fingerprint database for localization. As the fingerprints alleviate the multi-path problem and characterize the RSS property of each indoor location, SpiLoc achieves a fine-grained localization performance. We implemented the system and evaluated SpiLoc in a $45 \times 38m^2$ testbed. The evaluation shows that our system achieves an average localization error of 2.76m with low start-up and maintenance costs. Since SpiLoc requires no dedicated calibration and adaptively updates itself every time an RSS trace mapping is performed, it can be easily deployed to dynamic environments for fine-grained passive localization.

1.3.3 *A²Loc*: Accuracy Awareness of Wireless Indoor Localization

WiFi fingerprint-based indoor localization has been the focus of extensive research efforts [12, 88, 49, 75, 63, 86, 76, 74] due to its potential for deployment

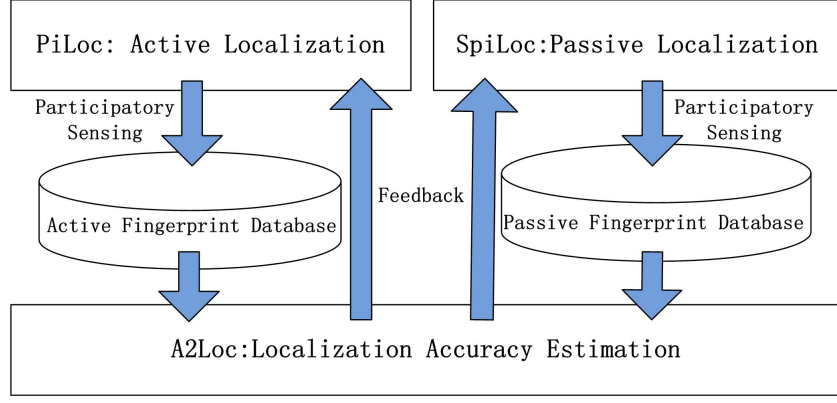


Figure 1.1: Overview of the works proposed in this thesis

without extensive infrastructure support. However, the accuracies of these different systems vary, and it is difficult to compare and evaluate these systems systematically. In most participatory sensing based indoor localization systems [63, 86, 76, 74], the radio maps can be automatically constructed and updated with significantly reduced calibration effort. However, there is currently no fool-proof way to measure the quality of the output radio maps directly. Without efficient approaches to provide direct feedback about the system accuracy, it is hard to judge the quality of the crowdsourcing data and decide how much data to use in the localization.

The accuracy awareness enabled by A^2Loc provides the ability to directly estimate the accuracy of the localization system over the area of interest. To achieve accuracy awareness, in A^2Loc we use a Gaussian process based approach that uses as input the radio map collected and localization algorithm to be evaluated, and outputs the expected accuracy of the system. A^2Loc is a set of algorithms to estimate the *point-level*, *region-level* and *floor-level* localization accuracies given the radio maps and localization algorithms used. In addition, useful information such as localization landmarks and the minimum number of sets of wireless access points required are also inferred directly. With efficient error-estimation algorithms, useful applications such as landmark detection, localization algorithm selection and access point subset selection are enabled.

In this work, as both PiLoc and SpiLoc leverage participatory sensing to output WiFi radio maps from the crowdsourcing input, A^2Loc acts as a complementary module that provides the accuracy feedback for both systems. As shown in Figure 1.1 above, the output of both PiLoc and SpiLoc can be directly

taken as the input of A^2Loc , which is then assessed based on their estimated accuracy level. Our evaluations show that A^2Loc provides efficient accuracy estimation and can serve as a useful tool for evaluation and performance tuning when developing fingerprint-based indoor localization systems.

1.4 Contributions

In summary, we make the following contributions in this thesis:

(1) We demonstrate that participatory sensing can significantly reduce the calibration effort for wireless indoor localization. By merging the crowdsourcing sensor data, the indoor floor plan can be automatically inferred and the radio maps required for localization are also built during this process. The self-calibrating capability of PiLoc enables minimum user effort for the bootstrapping and maintenance of active indoor localization systems.

(2) We show that fine-grained passive localization is possible using WiFi monitors with low start-up costs. The passive fingerprint database can be automatically inferred through crowdsourcing and statistical RSS trace mapping. Since SpiLoc requires no dedicated calibration and adaptively updates itself every time a RSS trace mapping is performed, it can be easily deployed to dynamic environments for fine-grained passive localization.

(3) We propose the introduction of accuracy awareness of wireless indoor localization. By taking the radio maps from arbitrary fingerprint-based wireless indoor localization systems as input, A^2Loc outputs the accuracy estimation and useful information such as landmarks that can be used to further improve the localization accuracy. A^2Loc makes systematic accuracy comparison feasible, and provides an efficient way for researchers to analyze the quality of the constructed radio maps either from dedicated site-surveys or participatory sensing. This capability makes it an efficient tool for evaluation and performance tuning for fingerprint-based indoor localization systems.

1.5 Thesis Structure

The rest of this thesis is structured as follows:

Chapter 2 provides a the literature review of the works that focus on wireless indoor localization and related research areas.

Chapter 3 presents PiLoc, a participatory sensing based active indoor localization system that calibrates itself using crowdsourcing data.

Chapter 4 presents SpiLoc, a passive indoor localization system that leverages the RSS trace mapping technique to efficiently bootstrap itself and provide fine-grained passive localization performance.

Chapter 5 describes A^2Loc , a set of techniques that gives direct accuracy estimations based on the output radio maps from wireless fingerprint-based localization systems.

Chapter 6 concludes this thesis by discussing possible directions for future work.

Chapter 2

Literature Review

In this chapter, we give an overview of the background and literature that is relevant to our work. We mainly cover the following topics: (1) active indoor localization; (2) passive indoor localization; (3) wireless signal modeling.

2.1 Active Indoor Localization

Smartphone indoor localization has received much attention recently due to the high demand from the industry and high commercial value of indoor location-based services (LBS), such as location-based advertisements and retail navigation. In the past two decades, active indoor localization has been the focus of a spectrum of research works. In active indoor localization, devices actively participate in the localization process to provide local information that can be used to infer the current location. Generally, these approaches can be categorized into five categories based on the system requirements and the underlying techniques used: infrastructure based, fingerprint based, propagation model based, SLAM based and participatory sensing based.

2.1.1 Infrastructure Based Localization

These systems rely on special-purpose infrastructures deployed to locate the target device. Early systems utilize short-range infrared [77] or RFID [57] and perform localization based on proximity. Cricket [62] uses radio and acoustic transmission and exploits the Time Difference of Arrival (TDoA) in the signals. Recent developments employ multiple-input, multiple-output (MIMO)

techniques using commodity APs and Angle of Arrival (AoA) to provide fine-grained localization [81]. While these techniques provide centimeter-level accuracy [81, 50, 62], the need for special-purpose infrastructure, the high deployment cost, and the infeasibility of localizing unmodified smartphones hinder their large-scale deployment.

2.1.2 Fingerprint Based Localization

A significant portion of research works on indoor localization explore the RF signal fingerprint-based approach. The basic idea is to fingerprint each location of interest and locate the device using nearest neighbor matching. The underlying assumption of this approach is that unique signatures can be found to fingerprint each location. The research for most of these works use WiFi RSS as the fingerprint [12, 88]. More recent works have proposed other forms of fingerprints, such as FM Radio [19] and physical layer information Channel Frequency Response [72]. SurroundSense [11] generalizes the concept of the fingerprint and explores ambient information such as noise, light color, etc. Fingerprint-based techniques reduce the deployment cost by leveraging the existing infrastructures and can achieve meter-level accuracy. However, these techniques suffer from high calibration costs, as a labor-intensive site-survey process is typically required in the offline phase to construct the fingerprint database (radio map) for each known location. The static radio map is also vulnerable to environmental dynamics, resulting in high level of maintenance. In this thesis, we aim to eliminate these overheads.

2.1.3 Propagation Model Based Localization

In trying to reduce the calibration effort, some researchers have proposed the signal propagation model based technique to estimate the RSS value at a given location based on the theoretic model instead of manually tagging [20, 48, 47]. One popular model is log-distance path loss (LDPL) [20], which estimates the RSS value based on the propagation distances. RADAR [12] also provides a model-based approach to estimate the RSS value based on the AP locations and floor plans. EZ [20] further improves this approach and only needs to measure the signal strength at a few locations. Compared with the fingerprint-based

techniques, model-based techniques typically reduce calibration effort at the cost of reduced accuracy. For most of these systems, AP locations or accurate floor plans need to be given.

2.1.4 SLAM Based Localization

Simultaneous Localization and Mapping (SLAM) techniques have been extensively studied by researchers in the robotic community. SLAM relies on landmark detection by camera, laser or other ranging sensors, and accurate controlled movement of robots. Several systems have been proposed to leverage the idea of SLAM by combining WiFi and IMU sensors on smartphones. Zee [63] exploits dead-reckoning and infers location according to the constraints imposed by the floor plan. However, it requires an accurate floor plan which is normally not available in practice. Combining user motion, SAIL [53] is able to achieve localization using a single access point.

2.1.5 Participatory Sensing Based Localization

To reduce the calibration effort, researchers have recently started to exploit participatory sensing to construct the fingerprint database in a more automatic way. The participatory sensing based scheme combines SLAM-based and fingerprints-based approaches. For example, UnLoc [76] exploits crowdsourcing and dead-reckoning to learn about indoor landmarks that exist in the environment to aid localization. However, it requires at least one ground truth location of the landmark. LiFS [86] exploits Multidimensional Scaling (MDS) to match fingerprints with an actual location using walking step information. These systems successfully reduce the effort in generating the radio maps, provided accurate indoor floor plans are given. Kim [36] proposes an autonomous fingerprinting method, but the method requires the strong assumption that the initial location and direction of the user are known a priori. Walkie-Markie [74] has recently proposed an algorithm to map pathways using WiFi-Marks. These systems rely either on accurate indoor floor plans or reliable landmarks that are present in the indoor environment.

2.2 Passive Indoor Localization

There is a growing interest in passive localization system, since they require no active participation of users or their devices. Many innovative applications are being developed to utilize the capability of passive localization. For example, the authors in [14] extracted social networks from smartphone probe messages, and analyzed the properties of the discovered social graphs, such as diameter, clustering coefficient and degree distribution. In [68], the authors propose analysis methods to extract temporal and spatial features from large sets of network-collected WiFi traces to better inform facility management and planning. In general, the passive localization techniques can be categorized as *device-free* and *device-based*.

2.2.1 Device-free Passive Localization

Device-free passive (DfP) localization [90, 70, 92, 83, 82] has been proposed to track entities without carrying any special devices. Most existing device-free passive localization systems rely on radio frequency (RF)-based techniques and the assumption that the existence or movement of human bodies will disturb the original RF patterns. In the location-based scheme [82], a passive radio map needs to be constructed in the calibration phase by recording the RSS measurements when a subject is located in each of the profiled locations. During the testing phase, the subject stands at any of these locations and the RSS matching is performed to infer the location of the user. In the link-based scheme [92, 61], however, the statistical relationship between the RSS measurements and the existence of the subject in the Line-of-sight (LoS) is measured, and the location of the user is inferred using geometric approaches.

Similarly, Radio Tomographic Imaging (RTI) based techniques [78, 79] try to reconstruct the tomographic image, and assume that the relationship between the location of the subject and the variations in RSS measurements can be mathematically modeled. Recently, MIMO radar-based techniques [10, 85, 9] have been proposed to track humans through analysis of body radio reflection. While these approaches do not require users to carry any device, the ability to track multiple entities simultaneously is still limited, and the systems are more vulnerable to multi-subject interferences.

2.2.2 Device-based Passive Localization

In device-based passive localization, devices attached to users are localized without active collaborations. With the increasing penetration of smartphones in recent years, users are increasingly carrying their smartphones all the time. Furthermore, with the proliferation of WiFi networks, the use of WiFi transmissions for passive tracking and monitoring of WiFi-enabled devices has recently gained much popularity [56, 14, 68]. Since each WiFi-enabled device transmits messages with a globally unique and persistent MAC address [60], smartphones have become trackable using WiFi monitoring equipment without the need of being connected to a specific WiFi access point or installing any apps. This is an important advantage over device-free passive localization, in which the number and identities of subjects being tracked are both hard to infer. Though smartphone manufacturers such as Apple have started to introduce features such as MAC randomization to smartphones from iOS 8, such features only work when the smartphones are not connected to the network and are in sleep mode [1]. Even with effective MAC randomization, there are still techniques for monitors to track the WiFi devices [1].

Several commercial systems are already on the market [6, 2]. Meshlium [6] detects any smartphone that works with WiFi or Bluetooth interfaces. The idea is to measure the number of people and cars that are present in a certain location (such as a shopping mall, an airports or a tourist attraction) at a specific time, allowing a study of the evolution of the traffic congestion of pedestrians and vehicles. The authors in [56] propose a passive coarse-grained outdoor tracking system for unmodified smartphones based on WiFi detection. A probabilistic trajectory estimation technique and some techniques for increasing the number of detected phones are described in [56]. However, *none of these systems achieve fine-grained passive localization*. In this thesis, we embrace the advantages of the device-based passive localization scheme, and propose a self-bootstrapped fine-grained localization system for smartphones. To the best of our knowledge, SpiLoc proposed here is the first passive indoor localization system that automatically constructs a passive fingerprint database and provides fine-grained localization performance.

2.3. Wireless Signal Modeling

System	Active/Passive	Category	Accuracy	Remarks
ArrayTrack [81]	Active	Infrastructure Based	< 0.5m	Additional infrastructure, does not work for smartphones
Ubicarse [39]	Active	Infrastructure Based	< 0.5m	Additional infrastructure, need to twist the devices
RADAR [12]	Active	Fingerprint Based	2~5m	Dedicated site survey
Horus [88]	Active	Fingerprint Based	~1m	Dedicated site survey
Zee [63]	Active	SLAM Based	1~3m	Requires accurate floor plan
SAIL [53]	Active	SLAM Based	~4m	Single access point, less accurate
EZ [20]	Active	Propagation Based	2~7m	No calibration, less accurate
UnLoc [76]	Active	Participatory Sensing Based	1~2m	Floor plan, seed landmarks
LiFS [86]	Active	Participatory Sensing Based	3~7m	Floor plan, less accurate
Walkie-Markie [74]	Active	Participatory Sensing Based	1~3m	Sufficient number of landmarks
Nuzzer [70]	Passive	Device-free	~2m	Dedicated site-survey, not suitable for tracking multiple objects
SCPL [82]	Passive	Device-free	1~2m	Dedicated site-survey, up to 4 objects
WiFi Tracking [56]	Passive	Device-based	~70m	Coarse-grained multi-device tracking

Table 2.1: State-of-the-art Indoor Localization Systems

2.3 Wireless Signal Modeling

To reduce the calibration effort for fingerprint-based localization systems, signal propagation models have been proposed in recent research works. A signal propagation model (e.g., the log-distance path loss (LDPL) [65]) can be used to predict the signal strength values at different locations in an indoor environment. RADAR [12] also employs a signal propagation approach to estimate the RSS value at various location, given the AP locations and the floor plan. [47] uses a zero-effort localization system that utilizes the RSS measurements made by APs to construct a model to map RSS to distance. These systems can predict the RSS value and reduce the calibration effort, but still rely on extending the capability of current off-the-shelf APs or the knowledge of AP placement, power settings, or floor plans. EZ [20] further reduces such requirements, and only needs to measure the signal strength at a few locations. While the proposed models provide insights into the signal propagation and the capability to predict the RSS values, the lack of uncertainty measurement makes them unsuitable for the purpose of accuracy measurement.

While [26, 84] also utilize a Gaussian process in the context of localization, they focus either on improving the localization performance, or the GP itself. Unlike all these existing methods, the accuracy awareness proposed in this thesis requires only the knowledge of the radio map and the localization algorithm used, and provides a direct assessment of the accuracy of fingerprint-based localization systems.

Chapter 3

PiLoc: Self-calibrating Active Indoor Localization

3.1 Introduction

Location is one of the most important types of context information in mobile and ubiquitous computing. Recently, indoor localization has been the focus of extensive research efforts [86, 76, 63, 10, 78, 79, 81, 50, 19, 11, 74, 25, 53, 93, 52, 39], due to both the need for indoor support of location-based services, and the unavailability of GPS in indoor environments. However, despite significant research progress, developing an indoor localization system that can be easily deployed on a large scale remains a challenge.

Two major obstacles hinder the large-scale deployment of such systems: (1) *Labor-intensive site surveys and system maintenance*: Many of these systems involve a dedicated offline calibration stage to build a radio map for the target location. The calibration requires the manual association of each location with its corresponding fingerprints, and needs to be repeated for any new locations. Furthermore, the radio map needs to be periodically updated to reflect the environmental dynamics. These dedicated and time-consuming calibration and maintenance efforts thus make these systems less practical for large-scale deployment. (2) *Lack of accurate floor plans*: Recent research developments [86, 63] have shown that the calibration effort can be reduced with the prior knowledge of accurate floor plans of the places being measured. However, accurate floor plans are often not easily available.

In this work, we attempt to answer the following question: *can we design an indoor localization system that can be easily deployed on a large scale?* Such a system should meet the following design goals. First, the system should not require specialized infrastructure support or prior knowledge of the environment, such as floor plans and locations of wireless Access Points (APs). Second, there should not be a need for an expensive manual-calibration or site-survey stage. Third, the system should be able to automatically adapt to environmental changes and require minimal maintenance-effort.

In this chapter, we propose *PiLoc*, an indoor localization system that calibrates itself through user-generated data. PiLoc is based on the following observations. First, sensor-enhanced smartphones are becoming increasingly pervasive. Second, a smartphone can record a user's movements (distance and direction), together with the names of APs within range and the associated signal strengths. Finally, it is possible to merge many walking segments annotated with displacement and signal strength information from users to derive a map of walking paths annotated with radio signal strengths. This last observation is central to the design of PiLoc.

By utilizing opportunistic sensing data contributed by users, PiLoc requires no prior knowledge about any building or any user intervention in both the calibration and maintenance stages. It adopts a novel trajectory matching and floor-plan construction algorithm to automatically cluster, filter, and merge all user inputs to automatically construct floor plans for different indoor areas. Most importantly, radio maps required for localization are also automatically built and updated in this process. PiLoc requires no special-purpose hardware, the only assumption in its use is the availability of a WiFi infrastructure.

3.2 PiLoc Active Indoor Localization System

3.2.1 Overview of PiLoc

The PiLoc architecture is shown in Figure 3.1 below. PiLoc exploits crowdsourcing to trace user walking trajectories using Inertial Measurement Unit (IMU) sensors installed in the smartphones. The IMU collects angular velocity and linear acceleration data, which are utilized as inputs to the system.

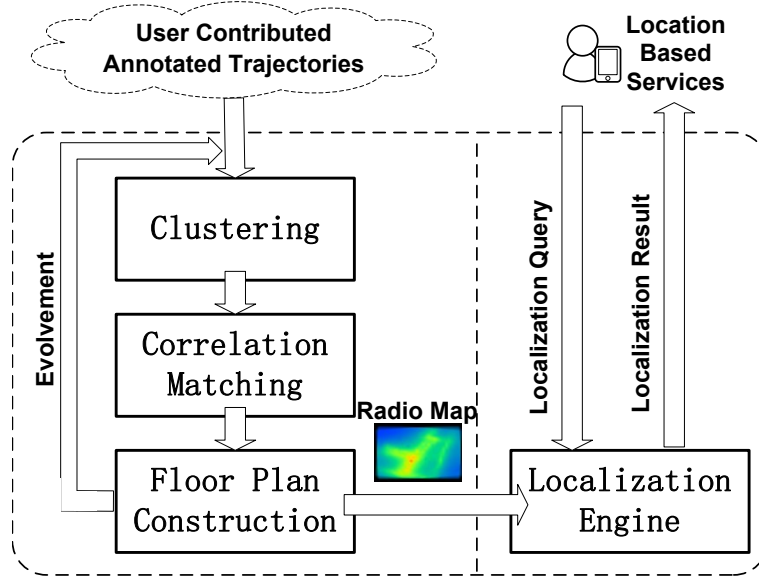


Figure 3.1: Overview of PiLoc

To enable localization, it is required that one or more users carrying smartphones with the data-collection application enabled walk on various parts of the indoor area to be localized, and upload the annotated walking trajectories collected. An annotated walking trajectory consists of discrete walking steps, which further consist of displacement vectors (distance and direction) and the WiFi fingerprints associated with the steps. There is no restriction on the walking patterns, and each walking trajectory can cover any part of the area. The limitation is that we can only localize areas that are covered by at least one walking trajectory, and localization accuracy improves with more trajectories. These user-contributed walking trajectories are used as inputs to construct or update the floor plan of the area covered by user movements.

The key challenge in PiLoc is how to combine these user-generated trajectories into a floor plan suitable for localization. There are three main steps involved. First, a clustering algorithm that uses AP signal strength and movement vectors is used to separate these walking trajectories into disjointed sets that cover different indoor floors and environments. In the second step, the system takes these disjointed segments and finds segments that match them based on movement vectors and AP signals. The matching is based on measurement of path and radio signal similarity between two different trajectory segments within the same cluster. Finally, in the third step, the system merges multiple trajectories to build floor plans. In the following sections, we present details of

these three steps.

3.2.2 Data Collection

3.2.2.1 Fingerprint Collection

Data collection does not have to be performed specifically for localization purposes. Instead, users equipped with smartphones walk around the targeted indoor environment as part of their daily activities. PiLoc opportunistically collects users' walking trajectories $T = \{\tau_i, i = 1, 2, \dots, m\}$. Each walking trajectory τ_i is determined by two stationary points detected by the phone's accelerometer. $\tau_i = \{s_1, s_2, \dots, s_n\}$, in which s_i is a discrete walking step detected by the linear accelerations from the corresponding phone accelerometer input. Besides stride length and heading direction, WiFi RSS fingerprints are also collected between every two consecutive steps, and are automatically associated with each step recorded. The heading direction of each step is obtained by converting the linear acceleration from the phone's coordinates to the world's coordinates. Therefore, each step $s_i = \{ID_i, x_i, y_i, f_i\}$ consists of four elements, global step identifier ID_i , horizontal displacement x_i , vertical displacement y_i and (radio) fingerprints f_i . 2D displacements x_i and y_i are calculated based on the headings (angle relative to the earth's North) and stride lengths, to identify the relative physical 2D position of the current step with respect to the first step s_1 in the same trajectory. For fingerprints $f_i = \{r_1, r_2, \dots, r_k\}$ represents the WiFi RSS measured at step i , where r_j is the received signal strength of the detected AP_j .

After collecting sufficient walking trajectories marked with corresponding fingerprints, PiLoc is able to construct floor plans and radio maps for the covered area. The speed of data collection is capped by the typical human walking speed. If we consider an indoor area with 100 meters of walk way and an average walking speed of four km/h, we can over one kilometer in 15 minutes or the entire walkway of 100 meters ten times.

3.2.2.2 Inertial Sensing

Dead-reckoning with smartphones has been explored in several previous works [26, 86, 76, 74, 63]. One significant challenge associated with dead-reckoning is the accumulated error over time. Therefore, dead reckoning can only be used

to track the user for a short period of time, otherwise, errors will need to be corrected frequently. This problem makes it very challenging to align and merge different user traces, especially in the construction of floor plans. This is also a major challenge for PiLoc. Several research works have been conducted to improve the accuracy of dead-reckoning with arbitrary phone placements [46, 63, 40]. Walking steps can be efficiently detected using a threshold-based sliding window algorithm [31]. In our experience, step detection is very accurate, and most of the time we can maintain exact step counts even after several hundred steps. Heading angles can be inferred by combining linear acceleration, compass, and gyroscope readings [46]. However, stride length varies for different users. In order to take this variation into account, we adopted the assumption from [63] that stride length follows Gaussian distribution, and used the default stride length with an additional 15% Gaussian noise.

As will be shown later, error in dead-reckoning is corrected in PiLoc by combining data from many trajectories in the merging process. In addition, outliers in the data will be filtered out via PiLoc’s merging and filtering process if these data do not match well with other data collected.

3.2.3 Trajectory Clustering

3.2.3.1 AP Clustering

As data collected from different users cover different parts of different locations, it is necessary to perform an initial level of data clustering to group the data into smaller, related groups. The goal of signal clustering is to divide all trajectories into geographically separated clusters. Each walking trajectory covers a particular indoor environment, and this clustering finds non-overlapping clusters based on the AP information. Given an input of n trajectories from all participating users, the AP clustering finds a clustering with l clusters $C = \{c_1, c_2, \dots, c_l\}$, such that:

$$\forall i \forall j \text{ } APSet(c_i) \cap APSet(c_j) = \emptyset, 1 \leq i \neq j \leq l \quad (3.1)$$

in which $APSet(c_i)$ returns the set of all APs that appear in at least one of the fingerprints in the trajectories of cluster c_i . AP clustering therefore separates trajectories collected in different indoor environments that have different sets

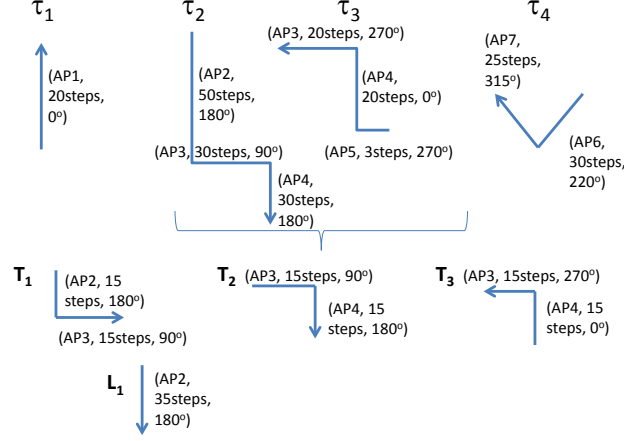


Figure 3.2: Examples of Trajectories and Clustering

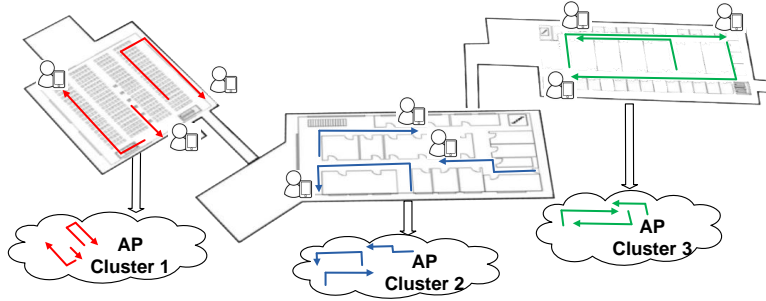


Figure 3.3: AP Clustering

of APs into different clusters. As an example, the four trajectories shown in Figure 3.2 below are separated into three clusters. The APs in each of the three clusters are $\{\tau_1\}$, $\{\tau_2, \tau_3\}$ and $\{\tau_4\}$. The corresponding set of APs are $\{AP1\}$, $\{AP2, AP3, AP4, AP5\}$ and $\{AP6, AP7\}$ respectively. As an illustration of the overall effect, as shown in Figure 3.3, the traces collected in three buildings are separated into three different clusters after AP clustering. Instead of relying on the fluctuating signal strength, AP clustering only detects the existence of APs, and provides a more reliable clustering. Though AP clustering only provides building-level granularity, this light-weight clustering is still an important technique to efficiently categorize the big trajectory data once the system is deployed at scale.

3.2.3.2 Floor Clustering

Floor Transition Detection. The trajectories collected from participating users cover different floors in different indoor buildings. The AP clustering pro-

vides an efficient way to distinguish disjointed indoor environments that have non-overlapping sets of access points.

To achieve floor-level clustering, we further annotate the walking trajectories with barometer sensor data. A barometer is a sensor that measures the surrounding air pressure. Pressure can in turn be translated into height above sea level (altitude) using the pressure-height equation [54]:

$$h = 44330 * \left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right) \quad (3.2)$$

where h is the altitude in meters, while p and p_0 are the measured air pressure and sea-level reference pressure, respectively, in millibars.

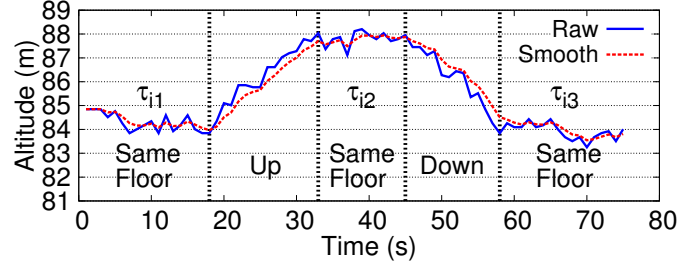
The dense altitude value provides a strong indicator of the floors from which the trajectories are collected. However, the altitude value calculated using Equation (3.2) is usually inaccurate without an appropriate sea-level reference from a nearby weather station. Therefore, we cannot directly use the absolute value of altitudes to determine the collecting floor of the trajectory. The measured relative change in height in the same trajectory, on the other hand, is very accurate [45, 69]. The barometer is sensitive enough to detect even the small change in height when a user travels from one floor to another.

Existing barometer chips have a noise value of less than a meter, making floor change detection possible [69]. Using a barometer is advantageous since it is inherently immune to phone position and usage. In addition, it is sufficient to sample a barometer at a low frequency, making the additional power consumption only a few milliwatts greater than for normal step detection.

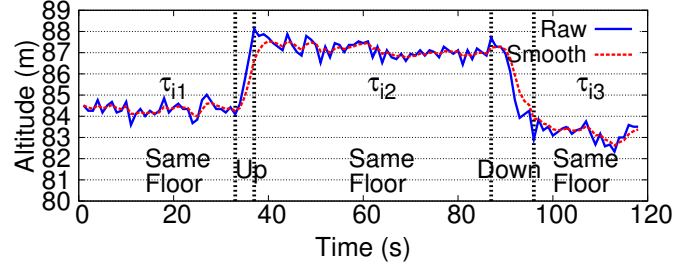
Figure 3.4 below shows how the altitude reported by the barometer changes when the user takes stairs and an elevator. When the user is walking on the same floor, the altitude remains stable. However, we can observe a marked change in height when the user is traveling up and down the stairs and elevator. We use this observation as the basis for accurate floor-transition detection in PiLoc.

We sample the barometer at a frequency of 1 Hz. To filter out the noise at the altitude detected by the barometer, we use the low pass filter:

$$h(t) = \alpha * h(t-1) + (1 - \alpha) * h \quad (3.3)$$



(a) Stairs



(b) Elevator

Figure 3.4: Altitude behaviors during different floor transition events. Floor transition separates trajectory τ into different floor segments

where $h(t-1)$ and $h(t)$ are the smooth altitude at time $t-1$ and t respectively, and h is the reported altitude by the barometer. In this work, α is set to 0.3 empirically. As shown in Figure 3.4, the low pass filter achieves smoothing altitude measurements while keeping the output responsive to altitude changes. To detect the floor transition, we maintain a sliding window of altitude values corresponding to steps taken by the user. For every new step taken by the user, we sample the barometer height and advance the sliding window by one step. If the difference in height between the end and start of the sliding window exceeds a threshold, we mark the event as a floor transition.

As illustrated by Figure 3.4, the floor transition splits each trajectory τ_i into different floor segments $\{\tau_{i1}, \tau_{i2}, \dots, \tau_{ik}\}$ if $k-1$ floor transitions are detected. To generate segments that cover only one single floor, we discard the parts of the trajectories during which the sliding window reports floor transitions. We do not know the exact floor from which the floor segments are taken, only that the two consecutive floor segments are taken from two different floors. For example, if the floor-transition detection algorithm reports that τ_{i1} has a mean altitude smaller than τ_{i2} , a floor transition constraint $\tau_{i1} \rightarrow \tau_{i2}$ is detected, which indicates that τ_{i2} was collected from a higher floor than that of τ_{i1} . Otherwise, the constraint

becomes $\tau_{i2} \rightarrow \tau_{i1}$.

The floor transitions impose constraints on the floor-level clustering process. We cannot infer the exact floor from which the trajectories are collected based on the absolute barometer readings, as the absolute value would vary with weather conditions. We use relative altitude values in PiLoc to detect floor transitions accurately. The accurate floor transition provides us with information on the segmentation point between two floors. We will demonstrate in the next section how we leverage this information to achieve floor-level clustering.

Floor-level Clustering. To cluster the collected trajectories into floor-based groups, we first need a similarity measurement for different trajectories. The similarity should be high for those collected from the same floor, and lower otherwise. Since the trajectories contributed by users are annotated with WiFi fingerprints during data collection, the floor-level similarity can be measured using the wireless signals collected. Different floors usually have different sets of WiFi access points. Even though there might be some overlaps in the AP sets, their signal strengths vary. The uniqueness of a WiFi fingerprint is also the fundamental assumption of any fingerprint-based indoor localization system. For two trajectories $\tau_1 = \{s_1, s_2, \dots, s_n\}$ and $\tau_2 = \{s_1, s_2, \dots, s_m\}$, the floor similarity $S_f(\tau_1, \tau_2)$ is defined as:

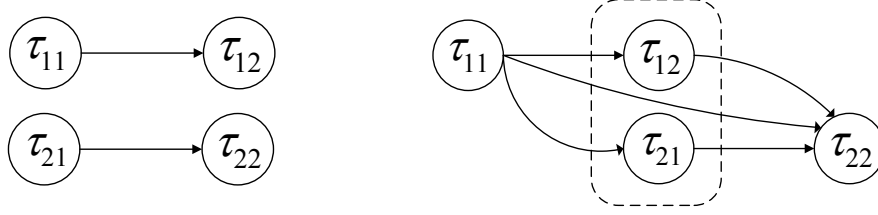
$$S_f(\tau_1, \tau_2) = \sum_{i=1}^n \sum_{j=1}^m S_s(s_i, s_j) / mn \quad (3.4)$$

where s_i and s_j are annotated steps in τ_1 and τ_2 respectively, and $S_s(s_i, s_j)$ is the fingerprint similarity of steps s_i and s_j using the Tanimoto Coefficient [22]:

$$S_s(s_i, s_j) = \frac{f_i \cdot f_j}{\|f_i\|^2 + \|f_j\|^2 - f_i \cdot f_j} \quad (3.5)$$

Here, f_i and f_j are fingerprints annotated to steps s_i and s_j respectively, as previously described. The fingerprint similarity between two steps $S_s(s_i, s_j)$ ranges from 0 to 1. The final output of floor similarity S_f combining all step similarities becomes the similarity metric between two trajectories and falls between 0 and 1 as well. If two trajectories have high floor similarity, they are more likely to have been collected from the same floor.

To illustrate the floor-level clustering process, consider a sample AP Cluster



(a) Floor Constraint Before Merging τ_{12} and τ_{21} (b) Constraint Update After Merging τ_{12} and τ_{21}

Figure 3.5: Floor Constrain Update

$c = \{\tau_1, \tau_2, \dots, \tau_{10}\}$ containing 10 trajectories. We do not know the exact floors from which they were collected, and trajectories in the same AP cluster might cover multiple floors. Based on the floor transition detection described in the previous section, we are able to detect those trajectories containing floor transitions. For example, if we have found a subset of five trajectories $c' = \{\tau_1, \tau_2, \dots, \tau_5\}$, such that each trajectory in c' contains floor-transition events, the floor transition detection will segment c' into $\{\tau_{11}, \tau_{12}, \tau_{21}, \tau_{22}, \dots, \tau_{51}, \tau_{52}\}$ if each trajectory contains only one floor transition.

Floor segmentation also generates a set of floor constraints $FC = \{\tau_{11} \rightarrow \tau_{12}, \tau_{21} \rightarrow \tau_{22}, \dots, \tau_{51} \rightarrow \tau_{52}\}$, if each trajectory is going upstairs in this example. Replacing the original trajectories in c with the newly generated floor segments, we obtain a new cluster $c'' = \{\tau_{11}, \tau_{12}, \tau_{21}, \tau_{22}, \dots, \tau_{51}, \tau_{52}, \tau_6, \dots, \tau_{10}\}$, in which each trajectory covers only one floor.

With the floor constraints we have, the goal of the floor-level clustering algorithm is to group these trajectories in c'' that were collected from the same floors into the corresponding floor clusters. Since the floor similarity between each pair of trajectories can be measured based on the wireless signal similarities using Equation (3.4), the clustering can be seen as a merging process to merge trajectories in c'' and generate disjointed floor clusters. Therefore the

floor clustering can be modeled as the following optimization problem:

$$\begin{cases} \text{maximize } \sum_i \sum_j S_f(\tau_i, \tau_j) \\ \text{s.t. } S_f(\tau_i, \tau_j) > t_0, \\ \text{Floor constrain } FC \end{cases} \quad (3.6)$$

where τ_i and τ_j are trajectories in c'' that are merged to the same floor cluster. The merging maximizes the sum of the floor similarities while ensuring that the floor constraint FC is not violated.

For each merged pair of trajectories, their floor similarity is ensured to be greater than the minimum similarity threshold t_0 . t_0 can be learned from each trajectory in c'' since each trajectory in c'' was collected from one single floor. To learn the average floor similarities for trajectories collected from the same floor, we split each trajectory in c'' evenly and calculate the average inter-similarity

Algorithm 1: Floor Clustering Algorithm

```

1 Input: AP cluster  $c$ 
2 Output: Set of floor clusters  $C_f = \{c_{f1}, c_{f2}, \dots, c_{fk}\}$ 
3 Generate  $c''$  with barometer-based floor-transition detection and generate initial
  floor constraints  $FC$ ;
4 Compute floor similarity  $S_f(\tau_i, \tau_j)$  for each pair of trajectories  $\tau_i$  and  $\tau_j$  in  $c''$ 
  using Equation (3.4);
5 Sort pairs  $(\tau_i, \tau_j)$  in descending order based on  $S_f(\tau_i, \tau_j)$ ;
6 for each pair of  $(\tau_i, \tau_j)$  do
7   if  $S_f(\tau_i, \tau_j) \geq t_0$  then
8     if  $\tau_i \rightarrow \tau_j \notin FC \ \&\& \ \tau_j \rightarrow \tau_i \notin FC$  then
9       if  $\tau_i$  or  $\tau_j$  not in  $C_f$  then
10        Merge  $\tau_i$  and  $\tau_j$  to the same floor cluster in  $C_f$ ;
11        Update floor constraint  $FC$ ;
12      end
13    else
14      if Clusters containing  $\tau_i$  and  $\tau_j$  can be merged based on  $FC$  then
15        Merge clusters containing  $\tau_i$  and  $\tau_j$ ;
16        Update floor constraint  $FC$ ;
17      end
18    end
19  end
20 end
21 else
22   return  $C_f$ ;
23 end
24 end
25 return  $C_f$ ;

```

between them using Equation (3.4). The minimum similarity t_0 is taken to be

the average floor similarity and we reject all those pairs with low similarities in the merging.

The floor constraints FC represent the knowledge that certain pairs of trajectories belong to distinct floors. Due to the transitivity of the floor constraints, they need to be updated in the merging process once we merge two trajectories into the same floor cluster. Consider floor constraints $FC = \{\tau_{11} \rightarrow \tau_{12}, \tau_{21} \rightarrow \tau_{22}\}$. As illustrated by Figure 3.5, if τ_{12} and τ_{21} are merged into the same floor based on their floor similarity in the merging process, the constraints need to be updated as $FC = \{\tau_{11} \rightarrow \tau_{12}, \tau_{21} \rightarrow \tau_{22}, \tau_{11} \rightarrow \tau_{21}, \tau_{11} \rightarrow \tau_{22}, \tau_{12} \rightarrow \tau_{22}\}$ due to their transitivity. The updating process must be performed whenever two trajectories are merged to the same floor.

The detailed steps of floor clustering algorithm is described in Algorithm 1. For each AP cluster c , the floor clustering algorithm finds a set of floor clusters that cover different floors of the indoor environment covered by this AP cluster. The barometer-based floor-transition detection first detects the floor transitions that are present in each walking trajectory and segments these trajectories to form c'' , in which each trajectory only covers one particular floor. The segmentation also generates the initial set of floor constraints FC .

To merge the trajectories in c'' , each pair of trajectories is first sorted by floor similarities in descending order. Each time, one pair of trajectories is picked from the top of the list. If their floor similarity is greater than t_0 and they meet the floor constraints, the trajectories become candidates to be merged to the same floor cluster. If one of these two trajectories does not belong to any existing floor cluster, both trajectories are merged to the same floor cluster, and FC is also updated due to the transitivity of the floor constraints. However, if two trajectories already belong to different floor clusters, we need to ascertain whether these two clusters can be merged. In PiLoc, if the average floor similarity of these two clusters is greater than t_0 and the merging will not cause any violation of the floor constraints, they are merged to the same floor cluster. Otherwise, we continue without updating the exiting floor clusters. The process is repeated until no such pair of trajectories can be found.

The resultant clusters consist of disjointed groups of trajectories, with each group covering one particular floor in this indoor environment. In PiLoc, the

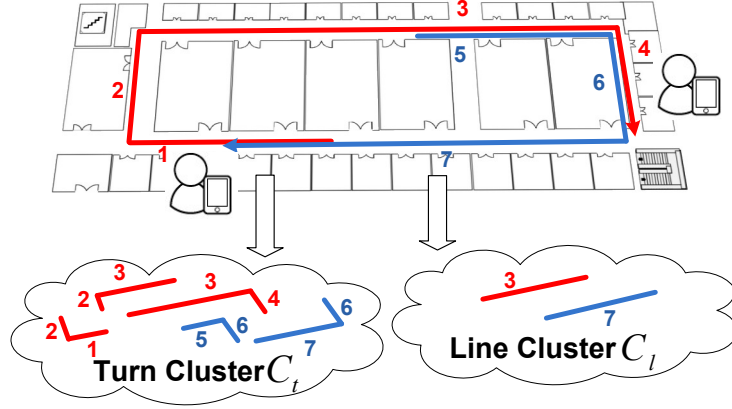


Figure 3.6: Path Segment Clustering

floor clustering algorithm is applied to efficiently generate a fine-grained clustering on top of each AP cluster.

3.2.3.3 Path Segment Clustering

Within the same floor cluster, we further divide a single trajectory into disjointed path segments. While path segments can take any form in general, in this work, we consider only two kinds of path segments, namely *turns* and *long straight lines*. Walking along a straight path and making corner turns are natural walking patterns in an indoor environment. A given trajectory $\tau = \{s_1, s_2, \dots, s_n\}$, can be broken into disjointed path segments (consisting of turns and/or straight lines) $S = \{s_p, s_{p+1}, \dots, s_q\}$ where $1 \leq p < q \leq n$. In dividing the trajectory, we first extract turns with minimum 5 and maximum 15 steps before and after the turning. After that, straight line paths containing more than 30 steps are extracted. As an example, consider the cluster consisting of τ_2 and τ_3 shown in Figure 3.2. Only three turns, T_1 , T_2 and T_3 are extracted. The fourth corner is not considered since the path before the turn is too short (fewer than five steps). Similarly, there is only one straight line segment (where AP2 is recorded). All other straight path segments are too short after the turn segments are removed.

We extract these segments from each trajectory and build third-level clusters $C' = \{c_t, c_l\}$ for each floor cluster in C_f based on path segments, where c_t is the cluster for turns and c_l is the cluster for long straight line segments. After second level clustering, each cluster c_t and c_l contains segments of the same path shape from the same indoor environment. Each segment S in c_t or c_l becomes the basic unit for trajectory matching in the next stage. The overall effect is shown in

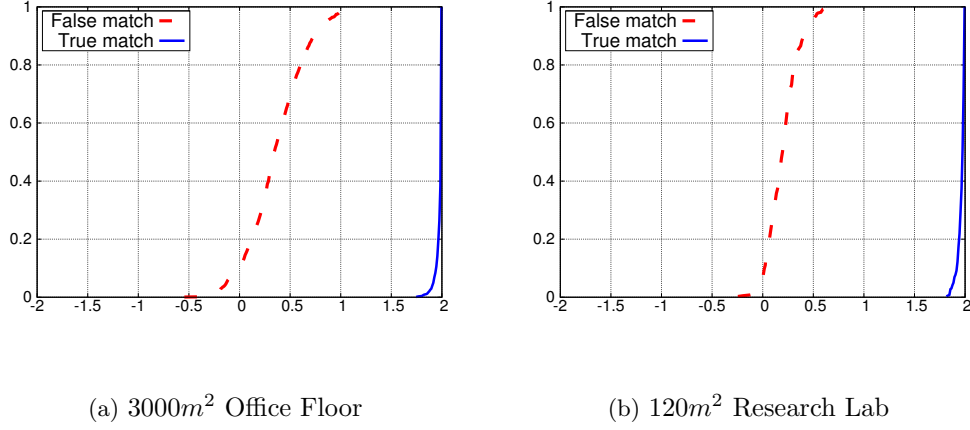


Figure 3.7: CDF of Path Correlation

Figure 3.6.

3.2.4 Trajectory Matching

A key difference between PiLoc and prior systems is that instead of using a WiFi signal or ambient information as landmarks, we utilize movement displacement (distance and direction) and the associated signal to match different segments. We have found that these parameters can provide high discriminative power for both dead-reckoning error correction and trajectory matching.

3.2.4.1 Path Correlation

Like the clustering component, the trajectory matching algorithm follows a two-phase scheme. The first phase is based on a simple but effective idea: when people walk along the same segment (turns or straight lines), the evolutions of the two trajectories on a 2D plane should be highly correlated. The path correlation correction can be measured as:

$$Corr_{path} = Corr_x(S_1, S_2) + Corr_y(S_1, S_2) \quad (3.7)$$

For two path segments from the same cluster c_t or c_l , $S_1 = \{s_1, s_2, \dots, s_n\}$ and $S_2 = \{s'_1, s'_2, \dots, s'_n\}$ with the same number of steps n , the Pearson correlation

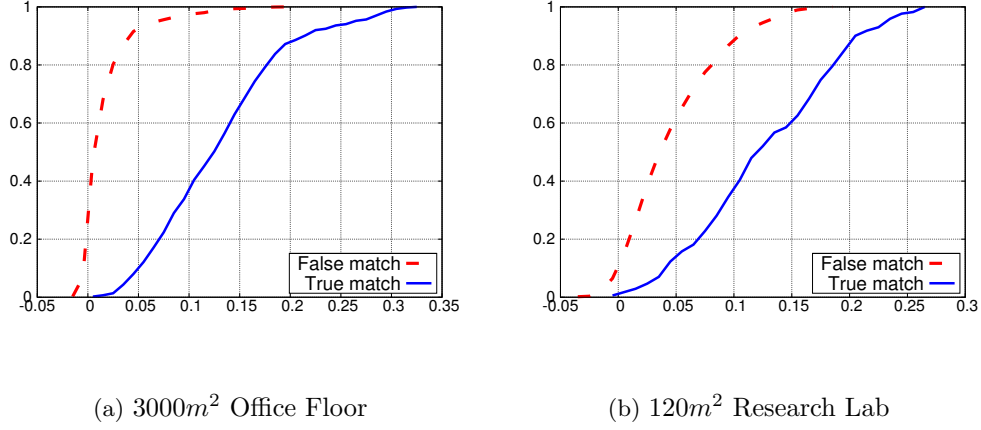


Figure 3.8: CDF of Signal Correlation

can be computed as:

$$Corr_x(S_1, S_2) = \frac{E[(X_1 - \mu_{X_1})(X_2 - \mu_{X_2})]}{\sigma_{X_1} \sigma_{X_2}} \quad (3.8)$$

where $X_1 = \{x_1, x_2, \dots, x_n\}$ and $X_2 = \{x'_1, x'_2, \dots, x'_n\}$ are the sequences of horizontal displacement of the steps of S_1 and S_2 , respectively. Similarly, $Corr_y$ is the correlation of the vertical displacements of the steps of S_1 and S_2 . These displacements can be computed given the step distance and direction of movement. $Corr_{path}$ therefore measures the similarity between two walking paths on the 2D plane.

Figure 3.7 shows the CDF of the path correlations for traces collected from both a large indoor floor level covering 3000 m² and a research lab covering only 120 m². Since one can walk along the same path in two directions, we computed the $Corr_{path}$ in both directions and took the higher of the two as the final path correlation. In both environments, more than 90% of path correlations for correct matches (paths with the same evolution trend on a 2D plane) have values greater than 1.90 (maximum 2). The path correlations are much lower for incorrect matches, with 90% less than 0.75.

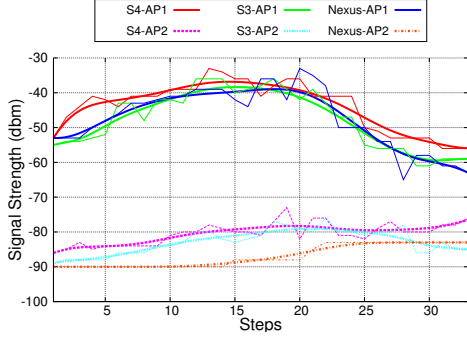


Figure 3.9: Stability of Signal Trends (Phone Varying)

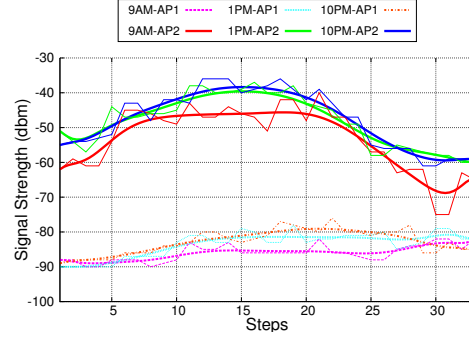


Figure 3.10: Stability of Signal Trends (Time Varying)

3.2.4.2 Signal Correlation

Path correlation alone is not sufficient for obtaining accurate matches. When path segments are collected from parallel corridors in the same building, these segments may have high path correlations. Another feature exploited in PiLoc is changes in the RSS signal along the walking path. It has been observed that an RSS signal changes according to a specific pattern along the same path-way. This change is due to the signal propagation and other environmental obstacles. The pattern according to which the RSS signal changes provides another useful hint to determine matching segments.

One uncertainty about using these signal measurements is the stability of their trends with respect to changes in phone model and time. Figure 3.9 shows the stability of WiFi signal trends on the same path across three different phone models (Samsung Galaxy S3, S4, and Galaxy Nexus). The trends are plotted with smoothed curves and are stable across different phone models for both APs. The variation is also relatively stable at different periods of the day. As shown in Figure 3.10, the RSS trends collected for the same walking path in the morning (9 a.m.), and afternoon (1 p.m.), and at night (10 p.m.) are also similar. Another observation is that the similarity between APs with higher RSS values tends to be higher than between those with lower RSS values. As shown in Figure 3.9 and Figure 3.10, the trend detected for AP1 is more stable than that for AP2. With these observations, we use signal correlation as a metric to further measure the similarity between two path segments S_1 and S_2 :

$$Corr_{signal} = \sum_i \omega_i \cdot Corr(R_1^i, R_2^i) \cdot I(R_1^i, R_2^i) \quad (3.9)$$

where $R_1^i = \{r_1, r_2, \dots, r_n\}$ and $R_2^i = \{r'_1, r'_2, \dots, r'_n\}$ are the sequences of RSS values of AP_i observed in S_1 and S_2 respectively. ω_i is the weight for AP_i and we set $\omega_i = \frac{2}{|\mu_{R_1^i} + \mu_{R_2^i}|}$. As signal strength values are given in negative terms (measured in dBm), APs with larger average RSS values will have more weight. $Corr(R_1^i, R_2^i)$ is the Pearson correlation of two RSS sequences for AP_i . $I(R_1^i, R_2^i)$ is an indicator function used to decide if an AP_i should be included in the computation.

$$I(R_1^i, R_2^i) = \begin{cases} 1, & |\mu_{R_1^i} - \mu_{R_2^i}| < \sigma_{RSS} \\ 0, & otherwise \end{cases} \quad (3.10a)$$

$$(3.10b)$$

where σ_{RSS} is the maximum acceptable difference between the two mean RSS values of two path segments. The current value for σ_{RSS} is set to 5 dBm, which has been observed to work well for different environments. As done with the path correlation computation, as movement can occur in both directions on the same path, we calculate the correlation for both the forward and reverse directions for each pair of segments, using the maximum correlation.

Note that not all APs are included in the computation. First, we exclude APs that appeared only in one segment and not in the other. Second, we also remove APs that appeared in fewer than 10 steps in either of the two segments. In summary, for the signal correlation computation, we only considered APs that appeared often enough in both segments, and whose average signal strengths are similar.

In general, the $Corr_{signal}$ increases as two trajectory segments have more common APs and the trends of the APs are similar. Figure 3.8 shows the signal correlation distribution for both the 3000 m^2 office floor and the 120 m^2 research lab. In both environments more than 42% of signal correlations for correct matches (same paths) have values greater than 0.15. The signal correlation is much lower for incorrect matches, with 98% less than 0.15.

3.2.4.3 Final Matching

PiLoc combines the discriminative power of both path and signal correlations in the final matching to achieve an accurate match. For each pair of segments in the

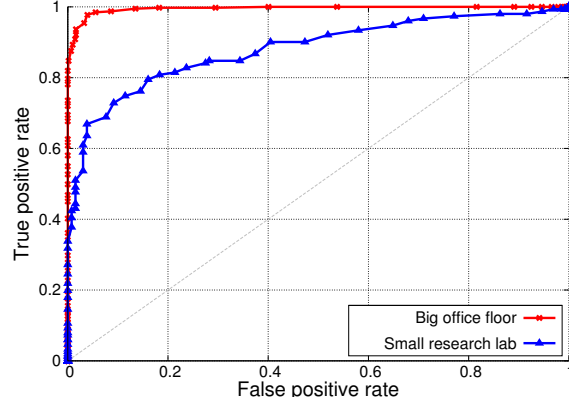


Figure 3.11: ROC Curve of Final Matching

cluster c_t or c_l , we first align them to have the same step numbers, and the turning point is used to align turn segments. In this way, PiLoc does not require that the starting and ending points of the path segments in the matching process be the same. We use path correlation threshold σ_{path} and signal correlation threshold σ_{signal} to find matching pairs.

In order to evaluate the accuracy of our matching algorithm, we have to obtain the ground truth of how the different segments matched through manual tagging. Figure 3.11 shows the receiver operating characteristic (ROC) curve for both the large office floor and the small research lab. Both curves show high levels of matching performance, with a large area under the curve. A good operating point can be chosen using the $y = x$ line. This operating point provides a guide for choosing the appropriate thresholds for the path and signal correlation values to be used for matching.

3.2.5 Floor Plan Construction

3.2.5.1 Algorithm

In PiLoc, the inaccuracy of the IMU and WiFi signal strength measurement makes it challenging to merge trajectories from different users. PiLoc addresses this challenge by merging and filtering all users inputs in the floor plan construction algorithm. The trajectory matching algorithm discussed in the previous section generates matching pairs for all segments from the same indoor environment. The output of the matching algorithm $\mathcal{M} = \{(S_1, S_2), \dots, (S_i, S_j)\}$ contains pairs of matched path segments and these matching pairs are used as inputs to the algorithm.

Algorithm 2: Floor Plan Construction Algorithm

```

1 Input: Matching result  $\mathcal{M}$ , Trajectories set  $T$  of 1 cluster  $c$ 
2 Output: Updated displacement matrix  $M_d$ 
3 Initialized displacement matrix  $M_d$ ;
4 for each matching segment pair  $(S_i, S_j)$  in  $\mathcal{M}$  do
5   // Collocate and determine displacements of
6   // matching steps
7   Set of collocated steps,  $S_{merge}$ , is initially empty;
8   for each matching step pair  $(s_m, s_n)$  in  $(S_i, S_j)$  do
9     Place  $s_m, s_n$  into a single location;
10    New displacement of  $s_m$  and  $s_n$  are average displacements of  $s_m$  and  $s_n$ 
11    to all points in  $S_{merge}$ ;
12     $S_{merge} = S_{merge} \cup s_n \cup s_m$ ;
13   end
14   for each step  $p$  in  $T$  or but not in  $S_{merge}$  do
15     Displacement of  $p$  = average displacements of  $p$  to all points in  $S_{merge}$ ;
16   end
17   Update displacement matrix  $M_d$  based on all new displacements calculated;
18 end
19 return  $M_d$ ;

```

Initialization. PiLoc merges and generates floor plans for all trajectories T collected in the same indoor environment, i.e., the same floor cluster c discussed in Section 3.2.3.1. In the initialization phase, PiLoc builds a displacement matrix M_d . Given two steps with global ID i and j , each belonging to one of the two matching segment pairs, the entry $M_d[i][j]$ gives the 2D displacement (x, y) between the positions indicated by the two steps as $(x_j - x_i, y_j - y_i)$. The displacement between two steps can only be measured if there are common matching path segments that can relate them. The displacement is “undefined” if the steps are from two different trajectories with no relationship.

Iteration. In the iteration phase, each matching segment pair (S_i, S_j) is taken into account to update the displacement matrix. Recall that matching segments have the same number of steps. For each pair of matching steps (s_m, s_n) , we “move” the starting position of these steps so that they start at the same point. We then compute the new displacements by finding the average displacements of these steps to those steps whose new displacements have been determined. The detailed steps of floor plan construction algorithm is described in Algorithm 2.

As an illustration, consider Figure 3.12. The trajectory consists of five steps $\{1, 2, 3, 4, 5\}$. $S_1 = \{1, 2\}$ and $S_2 = \{5, 4\}$ are the only pair of matching segments in this example. The algorithm first computes the starting (relative) position of

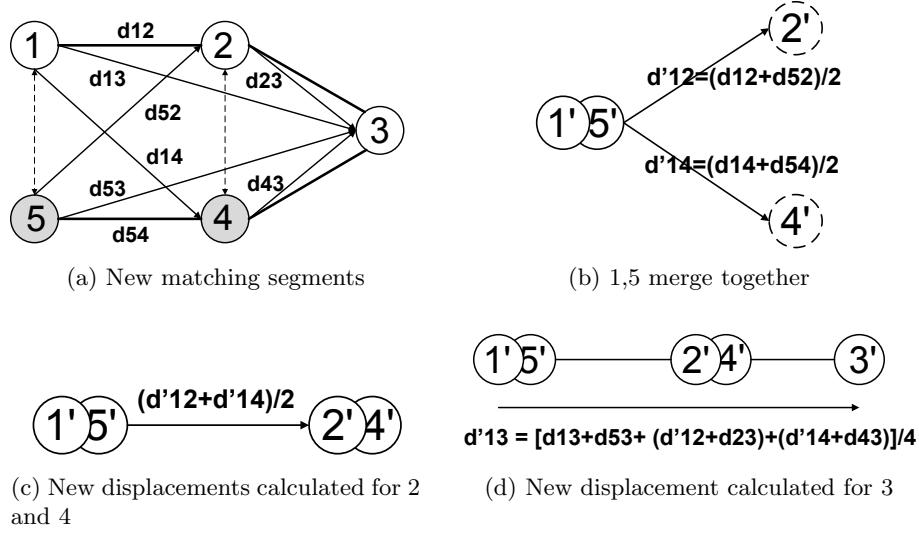


Figure 3.12: Example of Motion Vector Merging. d_{ij} denotes the current displacement and d'_{ij} denotes the new displacement.

the first matching steps. Figure 3.12(a) shows the original displacements of the points in the trajectory.

In Figure 3.12(b), the starting points of the first pair of matching steps $\{1, 5\}$ are considered to be at the same location (shown as $1'$ and $5'$ in the figure). In order to calculate the new displacements for the next pair of matching steps $\{2, 4\}$, which is again assumed to be collocated, the new displacements d'_{12} and d'_{14} are computed as $\frac{d_{12} + d_{52} + d_{14} + d_{54}}{4}$, as shown in Figure 3.12(c).

After the new displacements for all matching steps in this segment have been computed, the displacements of all the other steps are updated. As shown in Figure 3.12(d), the displacement d'_{13} is determined by averaging the displacements of all four matched steps.

Since the matching pair can be either from the same trajectory or different trajectories, the floor plan construction algorithm works for both intra-graph merging and inter-graphs merging. As shown in Figure 3.13, the trajectory is refined internally and merged with itself using the algorithm. The error cumulated in dead-reckoning is corrected using data within the same trajectory. Figure 3.14 shows the merging of different trajectories collected from the same floor. Note that since each step carries fingerprint data in the floor plan constructed, it naturally can serve as the radio map to handle localization queries and decide the current user location on the map. Since the merging algorithm works for all

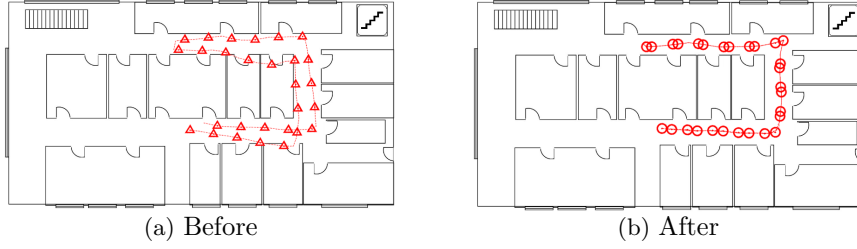


Figure 3.13: Intra Trajectory Merging

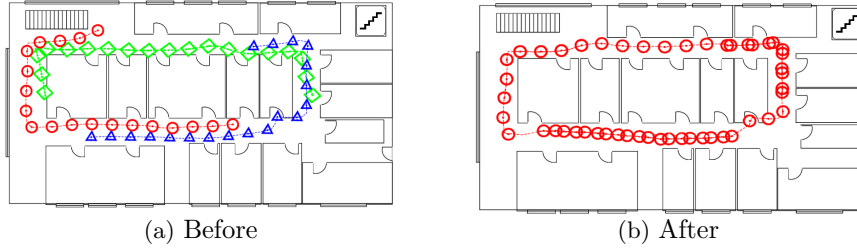


Figure 3.14: Inter Trajectory Merging

geographically separated clusters, floor plans and radio maps are generated for all different indoor environments covered by the participating users.

The maps generated are relative maps, i.e., the locations in the map are not associated with the absolute location yet. To map the floor plan to the real locations in the indoor environment, PiLoc only requires that at least one point be associated with a GPS coordinate. This point becomes a global reference point, and all the locations of rest of the points in the maps can be fixed.

3.2.5.2 Floor Plan Filtering

Filtering is required to remove the noisy samples and trajectories in the floor plan construction process. Trajectories that have no matching segments are first filtered out after the matching process. Therefore the outlier trajectories will not be reflected in the final results. To further smooth the constructed floor plans, we adopt a grid-based filtering scheme. The generated floor plans are divided into $1 \times 1m^2$ grids. We observed that most grids that contained correct walking trajectories have more steps than the average number of steps over all grids in the floor plans generated by the trajectory merging algorithm. In the final floor plan constructed, all grids with numbers of steps less than the averaged are removed. To smooth the floor plan constructed, morphological operators dilation and erosion [5] are used, and the extracted contours from the erosion

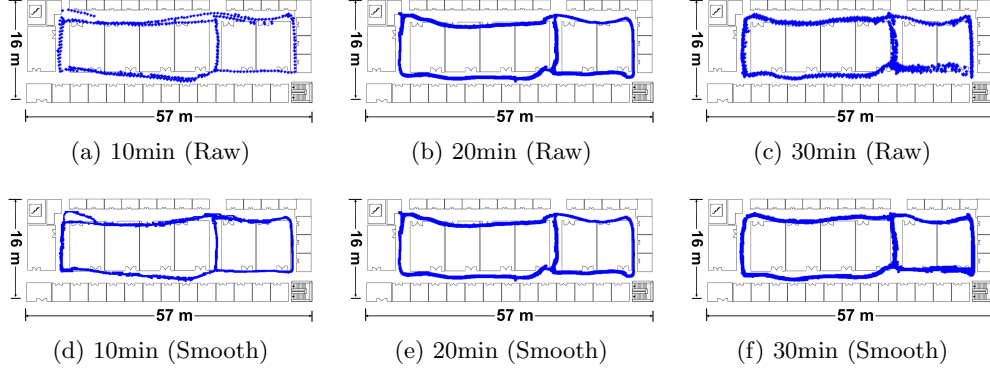


Figure 3.15: Floor Plan Evolution

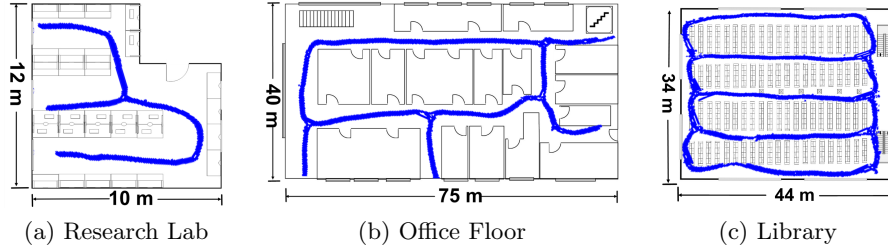


Figure 3.16: Floor Plan Construction for Various Indoor Environments

result are used as the smoothed walking paths.

3.2.5.3 Floor Plan Evolution

To reflect the environmental changes and new user inputs, the floor plan generated needs to be periodically updated. One important feature of PiLoc is that the floor plans will keep evolving with continuous incoming user inputs. The evolution is also fully automatic. In PiLoc, the floor plan is updated every 10 minutes to handle the new user input. All new data will be clustered into the existing clusters, or new clusters (e.g., new floors) may be generated. As shown in Figure 3.15, the floor plan is updated every 10 minutes to generate an evolving indoor map. The radio maps are also updated during the same process to maintain an up-to-date localization database.

3.2.5.4 PiLoc Localization

PiLoc adopts a fingerprint-based approach for indoor localization. The radio maps are automatically built and updated by merging user-contributed walking data. In this way, PiLoc is able to handle localization queries and return the

user location using the radio map and input fingerprints. Previous systems such as RADAR [12] utilize the fingerprint database by using the nearest neighbors from the query point to the reference points in the database as the similarity metric.

Such an approach works relatively well for indoor areas with sparse AP deployments (In RADAR, only three APs are presented). However, during our data collection we observed that many indoor environments have very dense AP deployments (more than 100 on one floor). Nearest neighbor matching works poorly at the dense AP environment, because at each location, smartphones can observe a long list of remote APs with RSS ranging from -80dbm to -90dbm. The RSS fluctuations of large numbers of these remote APs overwhelm the small set of nearby APs in calculating the similarity. However, nearby APs are more important in deciding the current location of the user since high RSS values only cover a small area for each AP. Based on this observation, PiLoc uses the simple but more effective *weighted maximum similarity* as the metric:

$$WMS = \sum_{i=1}^n \omega'_i \cdot \frac{1}{\max\{|r_i - r'_i|, 1\}} \quad (3.11)$$

where n is the total number of APs, and $\omega'_i = 1/|\mu_i|$ is the weight of the i th AP and is inverse to the absolute of its mean value. Therefore, nearby APs with higher average RSS values will have higher weights. r_i is the input RSS of AP_i and r'_i is from the radio map. WMS will have a higher value if the input point and reference have more common APs and the RSS differences for nearby APs are smaller. The location will be determined by the maximum WMS matching in the radio map. The PiLoc localization provides better accuracy than the conventional approach, especially in a dense AP environment.

3.2.6 Energy Management

3.2.6.1 WiFi Scanning Modes

Collection Mode. During data collection, it is important to increase the collected fingerprint density when users are walking indoors. To increase the fingerprint sampling rate, we only scan Channel 1 (2412MHz), 6 (2437MHz), and, 11 (2462MHz) during data collection. These channels do not overlap with the com-

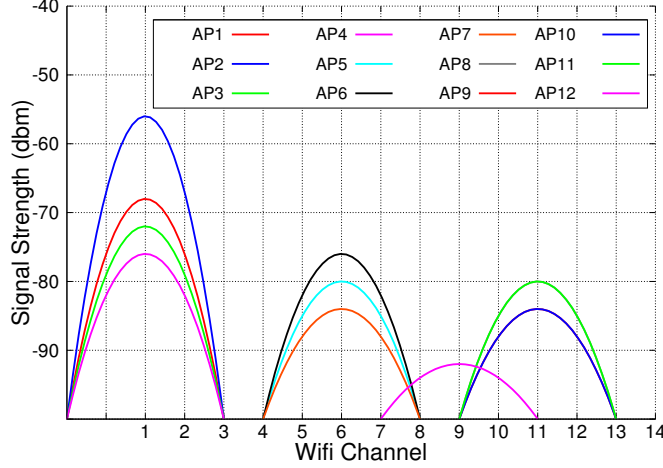


Figure 3.17: WiFi Signal Graph

monly deployed 802.11 b/g/n [59] network. As shown in Figure 3.17, these three channels covered most of the deployed APs in the environment we measured. In our scan, we also include one channel (5240MHz) from the less commonly deployed 802.11a network. By reducing the number of channels scanned and improving the efficiency of the code, we significantly increase the sampling rate. On average, around three radio fingerprints can be collected every second, compared with using the Android WifiManager which can only collect one sample every two to three seconds. The average number of fingerprints per step is computed by combining all fingerprints collected between two consecutive steps. However, the aggressive sampling also increases the energy consumption, and so needs to be performed as little as possible. We will discuss the sensor-triggered WiFi scanning scheme in Section 3.2.6.2.

Localization Mode. During online localization, the system becomes less sensitive to the WiFi sampling speed, and a two-to-three second WiFi refreshing rate is normally sufficient for most applications to achieve the 'real-time' localization. As a result, it is no longer necessary to sacrifice energy to WiFi sampling speed, and so we use the normal Android WifiManager scanning for online localization.

3.2.6.2 Sensor-triggered WiFi Scanning

To further reduce the power consumption of WiFi scanning, we exploit smartphone sensors to differentiate between different system states to switch the scan-

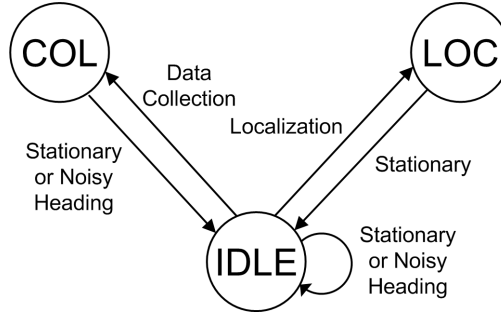


Figure 3.18: Sensor-triggered WiFi Scanning

ning mode dynamically. As shown in Figure 3.18, PiLoc runs in three scanning states: COL, LOC, and IDLE. In the COL state, PiLoc performs data collection and uses the fast scanning described in Section 3.2.6.1 to collect fingerprints as fast as possible. In the LOC state, PiLoc performs localization tasks and uses the normal Android WiFiManager scanning to reduce the sampling cost. In the IDLE state, PiLoc only samples the low-cost IMU sensors and stops all WiFi scanning to save energy.

Stationary Detection. During opportunistic data collection, as there is no control on participants’ walking patterns, they may stop occasionally. And when this occurs WiFi scanning will obtain duplicated fingerprints for the same location. Similarly, during localization, it becomes unnecessary to refresh the locations when users are staying at the same locations. To save power, it is important to reduce the WiFi sampling rate or stop WiFi scanning to avoid collecting redundant fingerprints for the same location. To detect when smartphone users are stationary, much research has been conducted to exploit the IMU sensors in phones [16, 35]. In PiLoc, as the system detects walking steps, the user is deemed stationary if the step counter is not updated for a given amount of time. In PiLoc, this period is set as 10 seconds. Users are determined to be stationary if no steps are detected within the waiting period.

Heading Noise Detection. PiLoc exploits opportunistic sensing to collect WiFi-annotated walking trajectories. Heading angle estimation using IMU sensors can be noisy [67] and the noise of heading angles calculated using smartphone IMU sensors constitutes a major error source of the system. In addition, users might put their phones in different places during data collection, for ex-

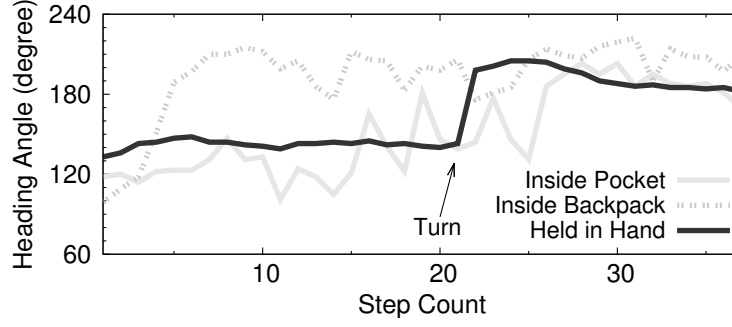


Figure 3.19: Heading Noise Detection

ample holding the phones in their hands, or putting them in pockets or backpacks. Although the trajectory merging process provides error correction for dead-reckoning as described in Section 3.2.5, it is important to filter out noisy compass readings before uploading them for merging. As shown in Figure 3.19, putting phones inside loose pockets or backpacks introduces more heading-angle fluctuations than when users are holding the phones in their hands during data collection. Detecting such noisy traces not only avoids adding additional noise to the trajectory merging process, but also provides important hints to the smartphones to switch to a low-power state to save energy.

In PiLoc, we opportunistically capture traces with smooth heading estimations and discard the rest. We measure the smoothness of the heading angles using the Hodrick-Prescott filter[29] to detect the level of fluctuation of the heading angles when walking:

$$Smoothness = \sum_{i=3}^n (\alpha_i - 2\alpha_{i-1} + \alpha_{i-2})^2 \quad (3.12)$$

where α_i is the heading angle sampled at the i th step. To keep detection real-time and robust, we maintain n as 10 steps and report heading noise when it exceeds an empirical threshold. The heading noise detection also triggers the smartphone to switch from the COL state to the IDLE state to save power.

Triggered Scanning. Figure 3.18 summarizes the state transition of PiLoc sensor-triggered WiFi scanning. During data collection, the smartphone will transit from the IDLE state to the COL state when the user is walking and the compass readings are not fluctuating, and will switch back to the IDLE state either when the user is detected to be stationary, or when noisy heading angles are

detected. Similarly, during localization the phone will switch to the LOC state from the IDLE state when the user is walking normally, and switch back to the IDLE state when the user stops walking. The detailed energy consumption of different states and the final triggered scanning scheme are evaluated in Section 3.3.3.5.

3.3 Performance Evaluation of PiLoc

3.3.1 Implementation

PiLoc has both client and server components. The client performs two functions: *data collection* and *issues localization query*. For data collection, the client runs an Android smartphone service in the background to opportunistically collect walking trajectories and radio fingerprints. For localization, the client issues queries to the server to localize the phone. The server collects user uploaded trajectory and fingerprint data. It uses the data collected to construct and update the floor plans periodically for all indoor environments it has data for. For each localization query, the server first determines the correct radio map to use based on the AP clustering result. The weighted maximum similarity match is then used to find the best matching location of the phone.

3.3.2 Data

The experimental data was collected over a one month period from five different areas which covered about 5800 m^2 in total. The layouts are shown in Figures 5.5 and 3.16. The sizes of these five different floors ranged from 120 m^2 to 3000 m^2 . The smallest area of 120 m^2 involved the inside of a research lab with lots of partitions, which posed a special challenge due to its very short turns and walk ways. Three different phone models are used: Google Galaxy Nexus, Samsung S3 and Samsung S4. All phones run the Android OS. An average of 37 APs are detected in each of the five areas. In total, 700 user trajectories are recorded, containing about 100,000 steps, with each step is associated with direction as well as WiFi fingerprints. In terms of time, these data corresponds to about 850 min of data collection.

3.3.3 Performance

3.3.3.1 Evaluation Metrics

We evaluate the overall performance of PiLoc by looking at the quality of the floor plan constructed and the localization accuracy. Two major metrics are used in the measurement for floor plan construction and localization:

- **Step Mapping Error (SME).** The floor plan constructed maps steps of walking trajectories into the real floor plan. The step mapping error measures how accurately the trajectories fit the real floor plan. Since fingerprints are associated with each step, a lower step mapping error results in higher fingerprint mapping accuracy, which directly affects the localization accuracy. The SME is defined as:

$$SME = \|L(s) - L(s')\| \quad (3.13)$$

where $L(s)$, $L(s')$ are the mapped location of the step and the ground truth location of the step respectively. A smaller SME reflects better matching of the constructed floor plan to the real one. To establish the ground truth, the locations where each step is taken in the reference floor plan are manually tagged. Since each step has a globally unique identifier, the location of one particular step in the constructed floor plan can be obtained by querying the ID, and SMEs are measured by calculating the differences between the estimated step locations and their respective ground truth locations.

- **Localization Error (LLE).** LLE measures how well the location given by the localization server matches the ground truth location of the phone.

$$LLE = \|L(p) - L(p')\| \quad (3.14)$$

where $L(p)$ is the estimated location and $L(p')$ is the real location of the phone. The smaller the Euclidean distance, the better the localization quality.

	Morning	Afternoon	Evening
Precision	100%	100%	96%
Recall	97.5%	98%	98%

Table 3.1: Performance of Barometer-based Floor-Transition Detection When Using Stairs

	Morning	Afternoon	Evening
Precision	89%	97%	91%
Recall	90%	89%	89%

Table 3.2: Performance of Barometer-based Floor-Transition Detection When Using Elevators

3.3.3.2 Trajectory Clustering

The clustering algorithms in PiLoc group user contributed data into smaller groups for higher efficiency in the later stage of floor plan construction process. Since the major uncertainty in the whole clustering process lies in the floor clustering process, we focus on the evaluation of floor clustering here.

Tables 3.1 and 3.2 show the measurement for the sliding-window-based floor-transition detection. The ground truth is input by the user whenever a floor transition occurs when the user is taking stairs or elevators. The collected time is also recorded for comparison. We group our data into different time period. We note that even when the barometer is sampled at a low sampling rate (1Hz), the floor transition can be accurately detected in all datasets. Since we use the relative altitude value instead of the absolute value for floor-transition detection in PiLoc, the accuracy remains high in all scenarios although the data were collected in different time periods. Floor transitions via stairs have above 96% precision, and above 97% recall. Similarly, for floor transitions via elevators, the average detection precision is 92% with average recall 90%. The relative altitude-based floor-transition detection in PiLoc makes it possible for robust detection from large quantities of input data that are collected from different users on different days.

To evaluate the floor clustering performance, we evaluate the quality of all generated floor clusters. If two trajectories clustered to the same floor cluster are actually from the same floor, this results in a true positive (TP), otherwise, it will be a false positive (FP). If the clustering algorithm groups two trajectories from the same floor into different floor clusters, will be a false negative (FN);

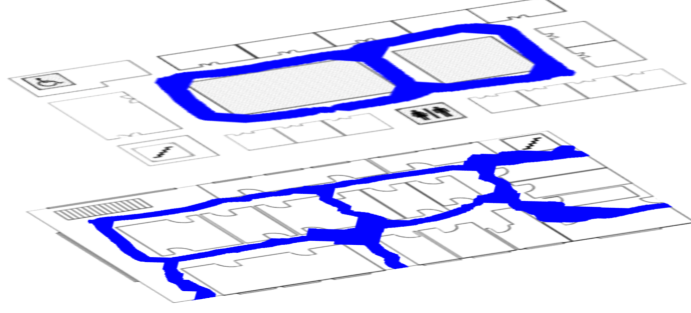


Figure 3.20: Multi-floor Floor Plan Construction

Precision	Recall	Accuracy
95.2%	88.9%	97.1%

Table 3.3: Floor Clustering Performance

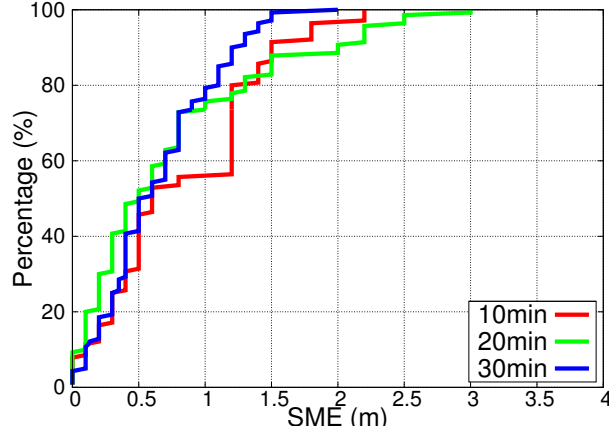
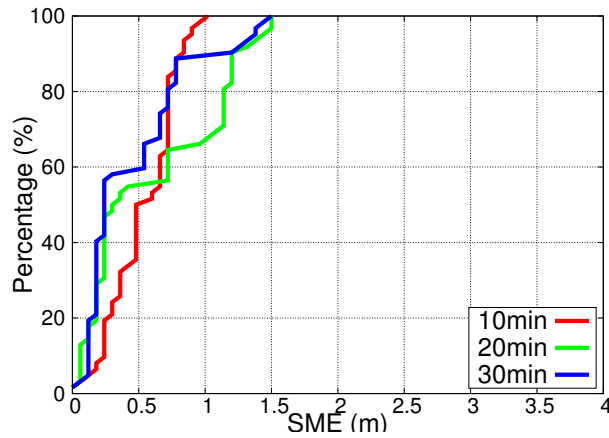
if not, a true negative (TN). In this way, we have $precision = TP/(TP + FP)$, $recall = TP/(TP + FN)$ and $accuracy = (TP + TN)/(TP + FP + TN + FN)$.

As shown in Table 3.3, the floor clustering algorithm using floor similarity and floor constraints can efficiently cluster trajectories into floor-based groups. The floor-clustering accuracy achieves an average precision of 95.2%, recall of 88.9%, and final accuracy of 97.1%. Since each floor cluster contains trajectories from a single floor, the floor plan construction algorithms can be applied to each individual cluster to generate a floor plan for that floor. By looking at the relative floor constraints obtained from all clusters, the relationships between each pair of floors can be obtained, resulting a multi-floor floor plan as shown in Figure 3.20.

3.3.3.3 Floor Plan Construction

To measure SME, each step associated with fingerprints is assigned a global ID. We tagged the ground truth localization for each collected step and measure the SME in the constructed floor plan. We plot the CDF for both the mid-sized ($900 m^2$) office floor and the $120 m^2$ research lab.

Figures 3.21 and 3.22 show three different CDF curves for the office floor and research lab respectively. Each CDF curve corresponds to a different time period of data collection, ranging from 10min to 30min. For the mid-sized office area shown in Figure 3.21, PiLoc achieves an average SME of 1.65m, 1.47m and 1.27m for 10min, 20min and 30min of data collection respectively.

Figure 3.21: CDF of SME ($900m^2$ Office Floor)Figure 3.22: CDF of SME ($120 m^2$ Research Lab)

For the research lab, PiLoc achieves an average SME of 0.54m, 0.6m and 0.46m for 10min, 20min and 30min of data collection respectively. Surprisingly, the accuracy for the research lab is better, probably because the step counting mechanism used incurs much less error for short distances.

3.3.3.4 Localization

Localization evaluation is performed for the large office floor ($3000 m^2$) and research lab. As shown in Figure 3.24, PiLoc achieves an average LLE of 1.37m for the research lab, with 80% of the errors less than 2.3m.

For the large office floor, the average LLE is 1.58m with 80% of the errors less than 3m.

Table 3.4 provides a brief summary of and qualitative comparison between PiLoc and other localization systems. As the evaluations are performed in different settings, the localization errors listed (obtained from the respective papers)

System	Average LLE	Effort
RADAR [12]	2~ 5m	Site survey
Horus [88]	~ 1m	Site survey
Zee [63]	1~ 3m	Floor plan
UnLoc [76]	1~ 2m	Floor plan, seed landmark
LiFS [86]	3~ 7m	Floor plan, less accurate
Walkie-Markie [74]	1~3m	Sufficient number of landmarks
PiLoc	1~ 3m	Does not rely on prior knowledge of indoor environment or landmarks, self-calibrating

Table 3.4: Listing of related localization systems

can only provide a high-level guide to the relative performances of the various systems. Even though PiLoc does not require manual calibration and landmarks, it can achieve localization accuracy that is comparable with that of the other localization schemes.

3.3.3.5 Power consumption

To evaluate the energy consumption, we use a Monsoon Power monitor to profile the power cost of PiLoc in three states. The one-minute snapshots for the different states are shown in Figure 3.23. We keep the display off for accurate measurement of all the three states. As shown in Table 3.5, the average power consumptions of the three WiFi scanning modes are 74.8 mW, 714.7 mW, and 852.2 mW. As shown in Figure 3.23, the COL state is the most power-hungry and incurs an additional 137.5 mW on top of the normal WiFi scanning used in the LOC state. Running PiLoc in both the COL state and LOC state incur roughly, additional power consumption of 700 mW more than in the IDLE state, when only IMU sensors are sampled, which indicates that we should switch to the IDLE state whenever possible. We simulate the state transitions of the sensor-triggered scanning scheme by looking at the step patterns and heading angles in the uploaded walking trajectories and measuring the final power consumption based on the percentage of time the system was in each state. As shown in Table 3.5, the sensor-triggered scanning reduces the average power consumption to 462mW, which corresponds to a battery lifetime of approximately 20 hours.

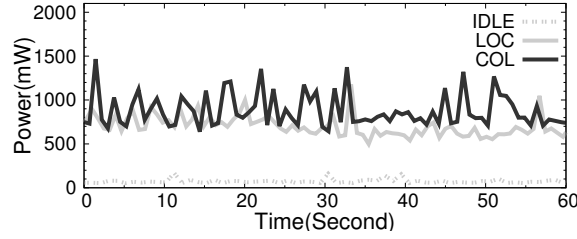


Figure 3.23: Power Profile of PiLoc in Different States

	IDLE	LOC	COL	Tri-Scan
Power	74.8mW	714.7mW	852.2mW	462mW

Table 3.5: Power Consumption Measurement

3.4 Discussions

3.4.1 Applications

Indoor localization plays a very important role in many real world applications. For example, location-based services and location-based advertisements have gained popularity. However, deploying and maintaining current indoor localization schemes requires too much effort, which hinders the development of location-based applications. By opportunistically collecting walking trajectories from causal users whose roles are not dedicated to localization, a localization system can be easily built and updated with PiLoc. For example, the movements of security guards or any other users can contribute traces for constructing the indoor floor plan of any given indoor environment. PiLoc provides an efficient way to leverage daily human movements for localization, and has the potential to be deployed on a large scale.

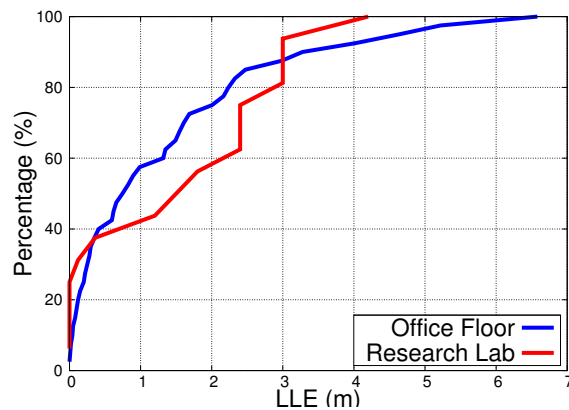


Figure 3.24: CDF of LLE

3.4.2 Limitations

PiLoc currently extracts turn segments and line segments for matching. Extending the system to more complicated layouts containing curve shapes requires the extracting of additional curve segments. In PiLoc, path correlation and signal correlation are used for trajectory matching to construct pathway floor plans. In open spaces where people may not walk along distinct walkways, path correlation and signal correlation may fail to differentiate intersecting or parallel aisles that are not separated by sufficiently large distances. This is one limitation of PiLoc. However, in practice walking paths inside buildings are often separated by walls or other obstacles. This will result in differences in signal correlations that can be distinguished by PiLoc.

3.4.3 Extensions

3.4.3.1 Diverse Floor Plans

In PiLoc, path segments are extracted and clustered for efficient matching. These path segments reflect the physical layouts of the floor plans. Although most indoor floor plans have rectangular layouts, some indoor layouts may contain curved walking paths. While a curved walking path may be captured as a series of straight lines and turns, the inaccuracy introduced can be substantial. Hence, to achieve a higher accuracy for these types of floor plans, we may have to include additional types of walking paths. Conceptually, adding additional path segment shapes in PiLoc is straight-forward, although the actual process of extracting these new shapes may be much more complex. Nevertheless, once the new paths are extracted, there is no change in the rest of the algorithms. The current architecture is thus highly extensible to diverse floor plans.

3.4.3.2 Enriching Constructed Floor Plans

While the localization system introduced in this work offers fast pathway floor plan construction and localization, this still does not constitute a complete indoor floor map. A complete indoor floor map should not only contain such a first-level skeleton structure, but should also contain an abundant number of elements that can be annotated into the path way floor plan. Such second-level elements can

be doors, stairs, escalators, elevators, or printers – items commonly encountered in an office building.

Such annotated elements can improve the indoor map in two ways. First, an enriched floor plan gives the user a better experience navigating through the area, via recognizing such human-oriented landmarks. Second, such elements also help to improve the localization accuracy of the indoor map. For instance, doors are important indoor indicators of changes in space, for example, entering one room from another. As important features in multi-floor buildings, stairs, escalators, and elevators are also useful in indoor navigation. Knowledge of their locations can therefore help a user decide a preferable direction and path to guide him to his destination.

3.4.3.3 Multiple Fingerprints

PiLoc utilizes WiFi fingerprints for localization. However, WiFi fingerprints are not tightly bound to our systems. Different fingerprints, such as FM radio signals [19] or even ambient noise [11], can be associated with each step and used in the localization phase. Also, to improve the performance, other fingerprints such as indoor magnetic fingerprints can also be added to the system to provide more information.

3.5 Summary

In this chapter we propose and evaluate PiLoc, an active indoor localization scheme that takes user walking trajectories as input and automatically builds and updates the indoor floor plan. By incorporating radio fingerprints, the indoor radio map is also automatically managed by PiLoc. PiLoc requires no human intervention and can achieve high localization accuracy with an average error of 1.5 meters. As PiLoc only requires minimal user effort for calibration and maintenance, it has the potential for large scale deployment.

Chapter 4

SpiLoc: Self-calibrating Passive Indoor Localization

4.1 Introduction

Indoor Localization systems such as PiLoc, as proposed in the previous chapter, achieve localization by relying on the cooperation of devices, and are usually referred to as *active localization*. Active localization is required by many applications such as user navigation, where users are willing to participate in the localization process. Recently, a new spectrum of applications that try to localize users without requiring their devices to cooperate explicitly have been developed. These applications include passive user tracking, customer-flow analysis, etc.

Recognizing these requirements, the research community has recently started to investigate *passive localization* techniques [56, 83, 14]. Compared with active localization, passive localization does not require the explicit participation of human or devices, and usually relies on the opportunistic overhearing of packets transmitted by smartphones [56]. Smartphones with WiFi interfaces enabled periodically send out messages even when they are not associated with any WiFi and even when the smartphone screens are off. This provides opportunities for WiFi monitoring devices to capture these transmissions and passively estimate the locations of the devices. Some previous work [56] has leveraged this idea by using WiFi monitors to track unmodified smartphones in an outdoor setting. While such work [56] illustrates the feasibility of passive tracking multiple smartphones, such tracking only achieves coarse-grained passive outdoor localization

Scheme	Category	Representative Systems	Remarks
Active	Infrastructure based	ArrayTrack [81], Ubicarse [39], COIN-GPS [58]	Requires explicit cooperation of device and relies on device information such as local WiFi scanning results and motion sensor data
	Fingerprint based	RADAR [12], Horus [88]	
	Propagation model based	EZ [20], Zero[47]	
	SLAM and crowdsourcing based	Zee [63], SAIL [53], UnLoc [76], LiFS [86], Walkie-Markie [74], MapCraft [80], PiLoc	
Passive	Device-free	Nuzzer [70], SCPL [82]	Infeasible to track multiple objects simultaneously
	Device-based	WiFi Tracking [56]	Coarse-grained localization performance
		SpiLoc	Goal: self-bootstrapping fine-grained passive localization

Table 4.1: Landscape of Indoor Localization Research

with a localization error of about 70 meters.

In this section, *we present our efforts to achieve fine-grained passive localization through self-bootstrapped passive fingerprinting using WiFi monitors.* Unlike with the RSS modeling used in [56], we choose to adopt the fingerprint-based approach due to the complexity of the RSS behavior caused by the multi-path effect [83] in an indoor environment. We propose *SpiLoc*, a self-bootstrapping passive indoor localization system that calibrates itself and provides fine-grained localization for smartphones. This system’s design was mainly based on the following observations. (1) With the knowledge of the indoor floor plan and the location of WiFi monitors, it is possible to opportunistically capture RSS traces that can be statistically mapped to specific indoor pathways. The mapping can be done even when smartphone transmissions are sparse. (2) By mapping the collected RSS traces from WiFi monitors to the walking paths, it is possible to bootstrap a passive fingerprint database for localization and achieve fine-grained localization performance.

Table 4.1 summarizes the current state-of-the-art indoor localization systems. Most of these systems belong to the active category, which requires the explicit cooperation of mobile devices. On the other hand, *SpiLoc* falls into the passive category, and has the following key differences:

- Unlike the active fingerprint based approaches [12, 88], *SpiLoc* relies on passive fingerprints. Instead of scanning WiFi beacons from mobile devices

actively, SpiLoc uses the signal strength measurement from deployed WiFi monitors when the signal-emitting devices are located at different indoor locations.

- SpiLoc has no control over the mobile devices and it is not possible to obtain local information such as inertial sensor data and local WiFi scanning results from the phone. Instead, the only information available is the RSS traces collected by WiFi monitors.

Due to these essential differences, SpiLoc has unique challenges:

- As there is no feedback from the mobile devices, motion related sensor data which is essential in active SLAM-based crowdsourcing solutions such as Zee [63], UnLoc [76], LiFS[86], Walkie-Markie [74], and PiLoc [52], is not available.
- Transmission rates from WiFi devices can differ widely, and the movements of these devices can also be highly irregular. Such behaviors further complicate the task of passive WiFi fingerprint crowdsourcing.

In SpiLoc, we use WiFi monitors to capture RSS traces from smartphones. Whenever two consecutive passive landmarks are identified from the RSS traces, we exploit the maximum likelihood based route inference technique to map the RSS traces to one walking path that connects the landmarks. After sufficient mappings are performed opportunistically, the passive fingerprint database is bootstrapped and the fine-grained locations of smartphones can be obtained in real-time.

4.2 SpiLoc Passive Indoor Localization System

4.2.1 Overview

4.2.1.1 System Architecture

The system architecture of SpiLoc is shown in Figure 4.1. At the beginning, the only knowledge the system has is the indoor floor plan and the locations of deployed WiFi monitors, which are typically available after system deployment [86]. The deployed WiFi monitors continuously collect the received signal strength (RSS) of WiFi transmissions from all smartphones that are in the vicinity. The RSS traces are then uploaded to a central server for both system

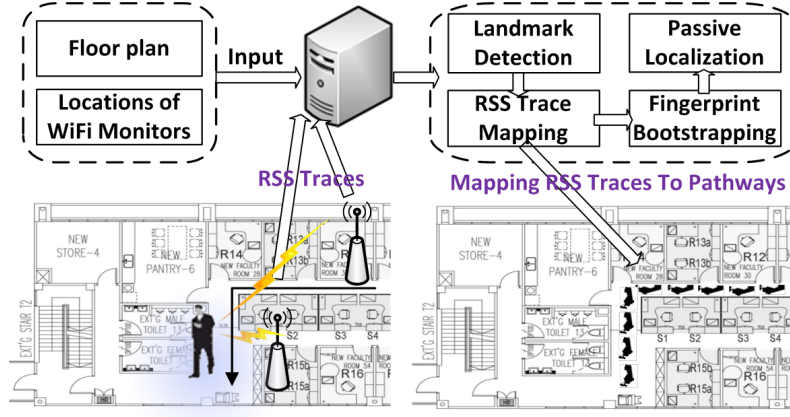


Figure 4.1: System Architecture

bootstrapping and real-time localization.

Passive landmarks are first detected from the RSS traces, which provide important information about a smartphone’s location at a given timestamp. Central to SpiLoc is the opportunistic trace mapping component, which opportunistically maps the collected RSS traces to one specific indoor pathway. For a particular user, once two consecutive passive landmarks are detected, the SpiLoc server performs the route inference to infer the most likely walking path that the smartphone user travels along, connecting these two landmarks. After the walking trajectory is estimated, SpiLoc maps the collected RSS from this user between these two landmarks to each of the locations in-between based on the data collection timestamps. After sufficient RSS traces are collected, the whole floor will be covered by the mapped RSS. Subsequently, all the mapped RSS measurements form the passive fingerprint database for this floor. Unlike the RF propagation model-based estimation, the constructed passive fingerprint database directly characterizes the RSS property at each indoor location and thus achieves fine-grained localization performance. With the bootstrapped fingerprint database, SpiLoc is able to handle online localization queries and achieve real-time localization given the RSS input from WiFi monitors, which is based on maximum likelihood estimation.

In SpiLoc, the passive fingerprint database updates periodically whenever a new trace mapping is successfully performed. The system therefore maintains an *evolving* RSS database of the floor and *adapts* to the environmental changes.

4.2.1.2 Opportunistic Data Collection

The WiFi monitors opportunistically overhear transmissions that are emitted by smartphones. It is known that smartphones periodically scan WiFi access points when they are not connected to the access points, which usually involves probe message transmissions [56]. When smartphones are connected to APs and have some ongoing tasks such as video streaming, they continuously send WiFi packets. Even when the smartphone screens are off, background services may also trigger wireless transmissions. All these transmissions from different smartphones are associated with their WiFi MAC addresses, allowing WiFi monitors to track the transmission traces of any individual smartphone that appears in the environment.

Let n denote the number of WiFi monitors that are deployed in an indoor environment. Assume a smartphone user is mobile. Each WiFi monitor captures the RSS of each transmission from the smartphone and generates traces $\tau = \{(t_1, r_1), (t_2, r_2), \dots, (t_k, r_k)\}$ for this phone, where r_k is the RSS value measured by the WiFi monitor at time t_k . Each (t, r) pair is recorded whenever a WiFi monitor captures one transmission from the smartphone. In SpiLoc, the distributed WiFi monitors in an area are synchronized, so timestamps can be used to merge the signal strength measurements from all monitors. Let $\{\tau_1, \tau_2, \dots, \tau_n\}$ represent the RSS traces continuously captured by these n WiFi monitors from the smartphone during a specific time period.

In the bootstrapping phase, after collecting enough RSS traces from all smartphones, SpiLoc opportunistically detects segments of traces that can be mapped to certain pathways to construct the passive fingerprint database. In the localization phase, RSS traces are used as inputs to localize smartphones in real-time, and these traces can also be used to update the passive fingerprint database. The core components of SpiLoc will be detailed in the following sections.

4.2.2 Passive Landmarks

4.2.2.1 Passive Landmarks: Concept

When bootstrapping the passive fingerprint database for localization, one key challenge is to associate the RSS traces captured by WiFi monitors with the

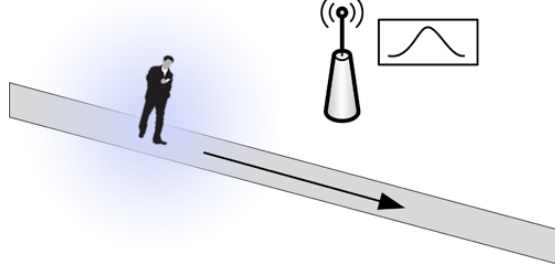


Figure 4.2: Passive RSS Trend

physical locations on the map. Previous work on WiFi-based localization [74] has used the *RSS trends* as WiFi-Marks to identify unique indoor locations. The key insight is that the WiFi RSS trends observed by walking users are normally stable for the same path, and the RSS tipping points in the trends can be identified as unique features of different locations.

This observation remains useful in the context of passive localization. As shown in Figure 4.2, as the smartphone user is walking past the WiFi monitor, the RSS of the smartphone transmissions captured by the WiFi monitor goes through an increasing phase, followed by a decreasing phase. Theoretically, the RSS tipping point corresponds to the closest location on the pathway in terms of signal propagation [74]. While RSS tipping points can be passively detected by WiFi monitors, similar tipping points can be detected when users are walking along different paths (e.g., parallel paths), which makes it unfeasible to uniquely determine the location of the user using the RSS trend alone. To address this problem, smartphone walking directions captured by IMU sensors are used to differentiate different RSS tipping points in [74]. In the context of passive localization, however, smartphone sensor reading is not available in the system.

To tackle this challenge, in SpiLoc, we combine the RSS trend with *RSS distribution* to opportunistically detect instances when users pass a location that is closest to the WiFi monitor. The RSS distribution is built over time for each WiFi monitor to uniquely characterize the signal strength distribution when users are in different indoor locations. Once RSS tipping points are detected from the trend and the RSS value falls in the highest part of the RSS distribution, the user can be traced to the location in the map that is closest to the WiFi monitor. SpiLoc uses such opportunistic detection to identify passive landmarks.

4.2.2.2 Passive Landmarks: Identification

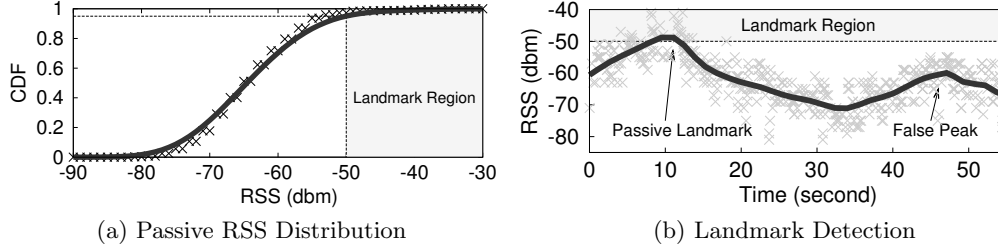


Figure 4.3: Passive Landmarks

Figure 4.3(a) shows the CDF distribution of RSS values detected from one WiFi monitor. The distribution captures the RSS property of all detected smartphone transmissions over time for this WiFi monitor, which is updated periodically to enable the system to gradually adapt to the environment. Since the majority of the RSS values are usually composed of transmissions when the smartphones are nearby, the RSS values outside the k -quantile of the CDF distribution are considered the *RSS landmark region*. For instance, the 95% quantile captures the top 5% of the RSS values.

As the WiFi monitoring is continuously recording the signal strength of smartphone transmissions, the RSS evolution trends can be measured directly. As shown in Figure 4.3(b), two RSS peaks are detected as the smartphone user is walking along the route shown in Figure 4.4. To validate the landmark, we pick the RSS values from each peak and see if they fall into the RSS landmark region. A passive landmark is detected only when both a clear RSS trend and high RSS values are observed, otherwise, the location of the peak cannot be determined, and the peak is marked as a false peak.

The detected passive landmarks provide important information that the smartphone users are located in the location closest to the WiFi monitor when an RSS peak is observed. In SpiLoc, we reset the location of the smartphone user whenever one passive landmark is opportunistically detected from the RSS traces, and use the detected landmarks in the trace mapping step.

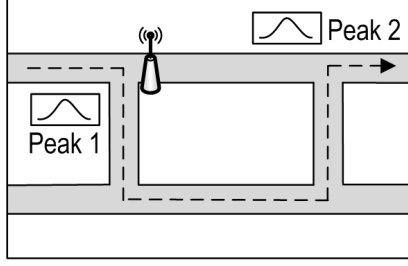


Figure 4.4: Different RSS Peaks When Walking Indoors

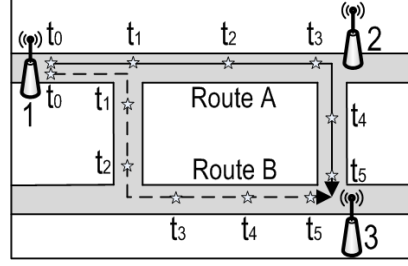


Figure 4.5: Route Generation Between Two Landmarks

4.2.3 Trace Mapping

4.2.3.1 Walking Route Inference

When sufficient landmarks are detected, it is possible to map the RSS traces collected between landmarks to the indoor pathways and construct the fingerprint map. Consider the example shown in Figure 4.5. Three WiFi monitors are deployed to record the RSS traces of smartphones continuously. From time t_0 to t_5 , each WiFi monitor records the RSS trace $\{(t_0, r_0), (t_1, r_1), \dots, (t_5, r_5)\}$. Assume two passive landmarks are detected at time t_0 and t_5 when users are walking past Monitor 1 and Monitor 3. If we can infer the correct walking route (either Route A or Route B in Figure 4.5) the user travels between two landmarks, we are able to map the RSS signals ($r_0 \sim r_5$) to the selected pathway based on their timestamps ($t_0 \sim t_5$), assuming the user travels at a consistent walking speed. Since there might be walking speed variations, we handle this problem through the variation detection technique that will be discussed later. Here we focus on the key challenge of accurate walking route inference, which maps the RSS traces to pathways.

For every two consecutive landmarks detected, the goal of trace mapping is to infer the correct walking route and map the RSS traces in-between to the inferred pathway. With the knowledge of the floor plan, SpiLoc first generates a set of candidate indoor walking routes. If the time taken to travel between two passive landmarks is relatively short, users usually tend to take the most direct walking route, which is usually contained in the k -shortest paths connecting these two landmarks. Therefore, in SpiLoc we use the k -shortest path algorithm [87] to generate the candidate route set $R = \{R_1, R_2, \dots, R_k\}$. Note that there is still a chance that the correct walking route is not included in the generated

candidate set, e.g., if it is cyclic route. Instead of trying to enumerate the infinite possibilities, SpiLoc exploits opportunistic mapping and handles the error introduced by false mappings with the noise filtering techniques discussed in Section 4.2.3.3.

To infer the most likely route that the user travels from the candidate route set R , SpiLoc leverages the *trend* of wireless signal cues. As the RSS is generally affected by the signal propagation distance, it is normally modeled with the log-distance path loss (LDPL) model [20]:

$$RSS_{ij} = p_0 - 10\gamma_i \log d_{ij} + \varepsilon \quad (4.1)$$

where RSS_{ij} is the measured RSS value of smartphone i by WiFi monitor j . p_0 is the RSS from smartphone i at a distance of one meter. γ_i is the rate of fall of the RSS, d_{ij} is the distance between the smartphone and the WiFi monitor, and ε is a random variable to capture the variations of the RSS measurements. Although the LDPL model is a theoretical model and the parameters need to be carefully trained to be accurate, the LDPL model provides important insights that we can leverage for the route inference. In SpiLoc, we do not rely on accurate RSS estimations from the model, but only leverage the relative RSS evolution trends revealed by the model. Figure 4.6 below compares the real RSS traces recorded by WiFi Monitor 2 with the theoretical RSS values calculated by the LDPL model, assuming users take different routes. It can be observed that even though the absolute value of the RSS calculated by the LDPL model is unreliable, the RSS evolution trend reflected by the model provides important hints about the route the user is traveling. In this case, the evolution trend of the real RSS trace matches the trend of Route A, and we therefore infer that Route A is the route taken by the user between the two landmarks.

To illustrate the route inference, consider the RSS trace $\{\tau_1, \tau_2, \tau_3\}$ collected by three WiFi monitors, as shown in Figure 4.5. Between two passive landmarks, each of the WiFi monitors captures six signal timestamp pairs $\{(t_0, r_0), (t_1, r_1), \dots, (t_5, r_5)\}$. To evaluate the likelihood of each candidate route R_j , we characterize the signal evolution trend of the real RSS measurements of monitor i using the *RSS evolution vector* $V_{ij} = (v_{01}, v_{02}, v_{03}, \dots, v_{mn})$, where v_{mn} is a binary value describing the change of RSS values between different times-

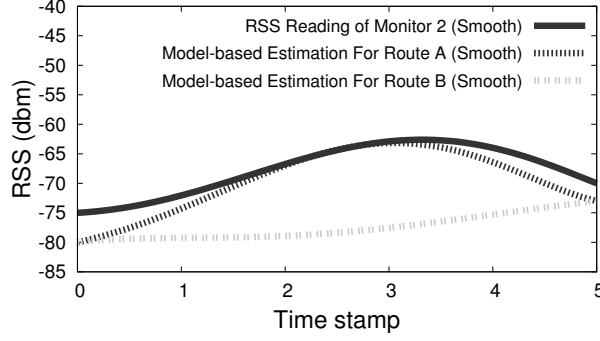


Figure 4.6: RSS Evolution Pattern Comparison

tamps t_m and t_n . Here, $v_{mn} = 1$ if $r_n - r_m > 0$, and $v_{mn} = 0$ otherwise. To address the fluctuations of signal strength in RSS measurements in order to estimate the trend correctly, we use the smoothed RSS values instead of directly using the raw RSS values captured from the WiFi monitor. Here, t_m and t_n are not necessary to be consecutive timestamps (i.e., $n - m \geq 1$). When timestamps t_m and t_n are farther apart, the physical distance between two RSS measurements in an indoor environment is usually larger, making the RSS change more obvious and useful in measuring the RSS evolution trend along the route. V_{ij} describes the signal increase/decrease patterns for each pair of different timestamps and is used as the ground truth RSS evolution pattern.

After the RSS signals are mapped to each of the locations along route R_j based on their collection timestamps, the theoretical RSS changes for monitor i can be modeled using the *model evolution vector* $V'_{ij} = (v'_{01}, v'_{02}, v'_{03}, \dots, v'_{mn})$, where $v'_{mn} = 1$ if the RSS values calculated by the LDPL model increase from timestamp t_m to t_n and vice versa. One advantage of comparing the relative RSS trends instead of the absolute RSS values using LDPL model is that the trends are parameter-free, and are only determined by the relative distances.

The differences between two vectors V_{ij} and V'_{ij} measure how the real RSS evolution measurement from monitor i matches the theoretical trend if the user is traveling along the selected candidate route R_j . For WiFi monitor i and its RSS traces τ_i , we use the normalized distance between two vectors to measure the likelihood of the candidate route R_j :

$$p(R_j|\tau_i) = \frac{\|V_{ij}\| - H(V_{ij}, V'_{ij})}{\|V_{ij}\|} \quad (4.2)$$

where $H(V_{ij}, V'_{ij})$ is the Hamming Distance between two vectors and $\|V_{ij}\|$ is

the number of elements in vector V_{ij} . The likelihood $p(R_j|\tau_i)$ for the selected route increases as the distance between the two vectors becomes smaller. Since we have n WiFi monitors ($n=3$ in this example), our objective is to find the route that maximizes the likelihood for all WiFi monitors. Therefore the route inference problem can be formulated as:

$$\arg \max_j \prod_{i=1}^n p(R_j|\tau_i) \quad (4.3)$$

The route inference in SpiLoc finds the route R_j from the candidate route set R that has the most consistent RSS evolution pattern with the theoretical model and maximizes the likelihood for all WiFi monitors.

4.2.3.2 Fingerprint Database Bootstrapping

Once the route connecting two passive landmarks is inferred, the RSS traces collected by each WiFi monitor are mapped to the corresponding locations along the route based on their timestamps, in order to bootstrap the passive fingerprint database. For example, in Figure 4.5, if Route A is inferred as the correct route and the time differences between all consecutive timestamps from t_0 to t_5 are the same, $r_0 \sim r_5$ will be evenly spread along the route with equal distances in-between, as shown in Figure 4.5. After sufficient trace mappings are performed, each indoor location will be covered by real RSS measurements. The mapped RSS measurements form the passive fingerprint for each location and the passive fingerprint database is bootstrapped for localization.

Since each WiFi monitor records RSS measurements from smartphones independently, all the mapped RSS values need to be merged to generate the fingerprint database. In SpiLoc, all WiFi monitors are synchronized by the Network Time Protocol (NTP), which provides millisecond time synchronization. Once an RSS trace between two passive landmarks is mapped to an inferred walking route, all signal timestamp pairs (t, r) in the RSS trace are combined based on their timestamps. For instance, (t_1, r_1) recorded by Monitor 1 and (t_2, r_2) by Monitor 2 are combined to generate an RSS vector (r_1, r_2) with the combined timestamp $(t_1 + t_2)/2$ if the difference between t_1 and t_2 is smaller than one second. After the traces from all WiFi monitors are merged, the fi-

nal RSS vector becomes (r_1, r_2, \dots, r_n) for n WiFi monitors, and r_n is set to *Nil* if Monitor n does not detect the smartphone during this period. The final combined RSS measurements (r_1, r_2, \dots, r_n) become the passive fingerprints, and are mapped along the route based on their combined timestamps. The fingerprints are then associated with their mapped locations and stored in the passive fingerprint database.

4.2.3.3 Noise Filtering

While the trace mapping in SpiLoc automatically bootstraps the passive fingerprint database for localization, false mappings inevitably introduce noise to the constructed fingerprint database. Noise filtering therefore becomes important in order to improve the quality of the fingerprint and the final localization accuracy. In SpiLoc, we leverage both *RSS trace filtering* and *fingerprint filtering* to improve system performance.

(1) RSS Trace Filtering

Temporal Filtering. Since the uncertainty of the route connected by two consecutive landmarks increases as their detection time difference increases, trace mapping becomes error-prone for those traces with large time differences between consecutive landmarks. To reduce uncertainty in the route inference process and avoid large amount of noise in the final constructed fingerprint database, it is desirable to filter out the RSS traces with large time differences before the route inference. In the implementation of SpiLoc, we only admit RSS traces for mapping if the time difference between two landmarks is less than one minute.

Walking Speed Variation Filtering. When mapping the RSS values onto the inferred walking route based on their timestamps, one important assumption is that users usually walk at constant speeds between two indoor landmarks. Although humans tend to walk regularly when they are walking continuously indoors, there are scenarios when the walking speed can significantly vary. For example, the walking speed will suddenly become zero when users meet their friends and stand still to have a conversation. The speed variation will significantly affect the fingerprint quality since RSS measurements are spread along the

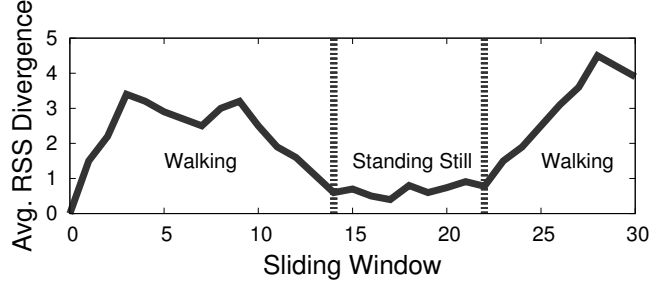


Figure 4.7: RSS Divergence Change with Walking-Speed Variation

selected route based on their timestamps, assuming a constant walking speed.

To address this problem, SpiLoc detects and filters out RSS traces with walking-speed variations by looking at their RSS patterns. One important intuition here is, if the user slows down or stands still, the RSS signals observed from all WiFi monitors usually stay similar for a period of time. Such an observation is an important indicator that the user is currently experiencing walking speed variation. To measure the RSS similarity, we maintain a window of ten RSS readings for all WiFi monitors. We measure the RSS divergence with $\sum_{i=1}^n Div(i)/n$, where $Div(i)$ is the standard derivation of RSS signals within the window of the WiFi monitor i . As shown in Figure 4.7, the RSS divergence becomes smaller when a user stands still, and increases accordingly as the user resumes walking normally. In SpiLoc, the RSS divergence is exploited to detect and filter out RSS traces with walking speed variations.

(2) Fingerprint Filtering

The route inference process in SpiLoc finds the most likely walking route in terms of RSS evolution trends. Although the mapping accuracy remains high, as we will show in the evaluation section, the false mappings introduce noise to the constructed fingerprint database. However, as the indoor floor is covered by dense mapped passive fingerprints, it becomes feasible to statistically filter out fingerprint noises that are present in the database to improve the quality of the fingerprints.

We treat each fingerprint FP as a multivariate random variable with n elements, where n is the number of WiFi monitors. To detect the fingerprint outliers, we evaluate the distance between one given fingerprint to the distribution of all nearby fingerprints. As each fingerprint is mapped to one physical

location in the indoor floor, all nearby fingerprints form a fingerprint distribution that characterizes the RSS properties of that region. In SpiLoc, we group all fingerprints within a distance of one meter to construct the fingerprint distribution. To measure the multivariate distance from one fingerprint to the distribution, we use Mahalanobis distance [23], which measures the distance from one fingerprint to the centroid of the distribution in multivariate space. The Mahalanobis distance $MD(FP)$ is calculated as follows:

$$MD(FP) = \sqrt{(FP - \mu)\Sigma^{-1}(FP - \mu)^T} \quad (4.4)$$

where μ is the mean vector and Σ is the covariance matrix. The distances are asymptotically chi-square distributed with n degrees of freedom (χ_n^2) [23]. Therefore, a multivariate fingerprint outlier can be determined if it has a large Mahalanobis distance. As a result, fingerprints that are a long distance from the centroid of the distribution in the multivariate space are filtered out as noise.

4.2.3.4 SpiLoc Localization

The filtered passive fingerprints are stored in the final fingerprint database for localization. For each sampled fingerprint $FP = (r_1, r_2, \dots, r_n)$ from n WiFi monitors in the localization phase, the goal of localization is to find the location x such that:

$$\arg \max_x p(FP|x) \quad (4.5)$$

Since the collected fingerprints are already mapped to different locations in the database, $p(FP|x)$ can be easily obtained by approximating a parametric distribution, such as a Gaussian Distribution, by combining all fingerprints at location x [89]. Since there might not be sufficient fingerprints at each location at the beginning, when the mapped RSS traces are sparse, we similarly combine all fingerprints within a one meter area at each location to approximate the fingerprint distribution at that region, in order to perform the final location inference using Equation (4.5).

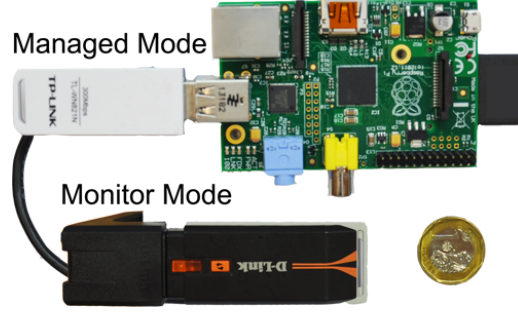


Figure 4.8: WiFi Monitor

4.3 Performance Evaluation of SpiLoc

4.3.1 System Implementation

The SpiLoc implementation consists of two components, *frontend WiFi monitors* and a *backend server*. As shown in Figure 4.8, each WiFi monitor consists of a Raspberry Pi, a D-Link wireless adapter (DWA-125) and a TPLink TL-WN821N wireless adapter. The DWA-125 is set to monitor mode to capture transmissions from all smartphones, and the TL-WN821N is set to managed mode to transmit real-time RSS traces to the backend server for system bootstrapping and real-time passive localization. The Raspberry Pi serves as a coordinator to control the RSS trace collection and backend transmissions, and also periodically synchronizes the local time with the network time using NTP.

The backend server receives all RSS traces and groups the RSS readings for each unique smartphone MAC address. As each WiFi monitor is synchronized with NTP, the time stamps in the RSS traces can be used directly on the server. The server keeps track of all wireless devices via their unique MAC addresses.

4.3.2 Evaluation

4.3.2.1 Experiment Design

Testbed. We performed our experiment on a 1710 m^2 indoor office floor. The layout of the floor is shown in Figure 4.9. In total, eight WiFi monitors are deployed. The location of each WiFi monitor is labeled in Figure 4.9. Each passive fingerprint therefore consists of eight elements, each of which corresponds to the RSS readings of the relevant WiFi monitor. The layout of the floor consists of 12 different turns, users walking around on the floor have different routes to travel

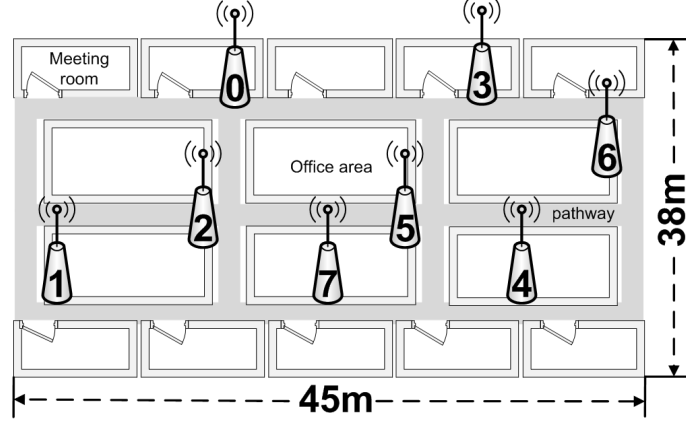


Figure 4.9: Layout of the Testbed

between each pair of WiFi monitors.

Data. To collect the data, we asked the participating users to walk randomly with smartphones on the floor. We collected RSS traces from all WiFi monitors as the smartphone users were walking. In total, about 300 minutes of RSS traces were collected. There was no restriction on the smartphone statuses: users could keep the phones in the idle state or perform background tasks when they were walking. To establish the ground truth, we asked the users to manually tap their locations on the map periodically. As the smartphones were synchronized as well, the time stamps and locations entered provided ground truth information about the routes they were traveling, and their physical locations. Since the walking speed would affect the mapping accuracy, we also asked the users to enter whether they walked at a fairly constant speed or undertook speed variation (such as standing still for a few seconds) during trace collection. We selected half of these traces for trace mapping evaluation and to bootstrap the fingerprint database, and used the other half to measure the final localization performance.

4.3.2.2 RSS Trace Mapping Performance

To test the RSS trace mapping performance, we extracted RSS traces between all consecutive landmarks detected. We performed route inference for each of the traces and evaluated the mapping accuracy by verifying the selected routes against the ground truth routes that the users traveled. To compare the mapping performance for different landmark pairs, we show the mapping accuracy for six different landmark pairs in Figure 4.11.

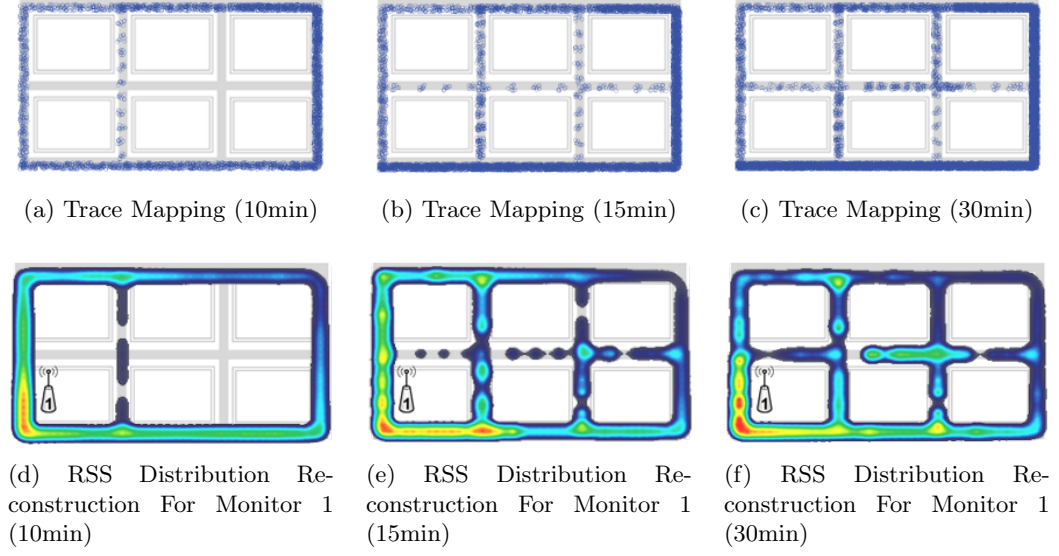


Figure 4.10: Trace Mapping and Fingerprint Database Bootstrapping

As shown in Figure 4.11, landmark pairs (1,0), (3,6), and (6,4) have the highest mapping accuracies for RSS traces without speed variation. Traces connecting these three landmark pairs are mapped to correct walking paths on the floor with more than 97% accuracy. The result indicates that if landmarks in each of these three pairs are detected consecutively, the route inference is able to infer the correct route with high accuracy. For these pairs, as the two landmarks are relatively close, in most cases the users traveled the shortest routes connecting the two landmarks. Otherwise, if the time taken between two landmarks is too long, there would have been a probability that the trace would have been filtered out in the temporal filtering process. Since the signal evolution patterns of all WiFi monitors for the short direct route appear unique compared with other longer routes, the accuracy remains high. For landmark pairs that are farther apart, e.g., (3,7) and (4,1), the mapping accuracies is relatively lower, with both about 70%. Overall, the route selection algorithm is able to efficiently map traces to the correct route if there is no speed variation in the traces. Figure 4.12 shows the CDF of the mapping accuracies of all landmark pairs, the average mapping accuracy for all different landmark pairs is 85.7% for the collected traces.

Figure 4.10 illustrates the evolution of indoor fingerprint coverage as different amounts of traces are collected. To demonstrate the evolution process of passive fingerprint mapping, we only show the trace mapping for the first 30 minutes

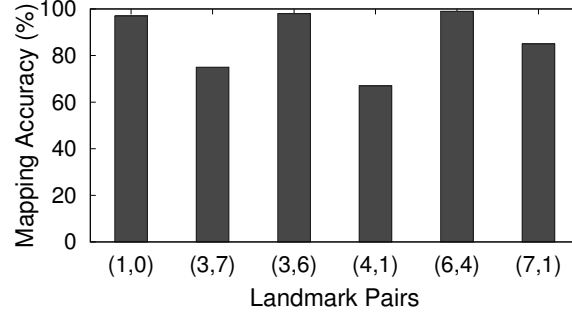


Figure 4.11: Trace Mapping Performance For Traces Without Walking Speed Variation

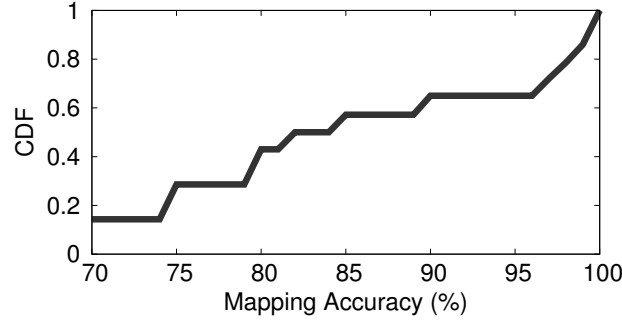


Figure 4.12: CDF of Mapping Accuracy For All Landmark Pairs

of collected RSS traces here. As shown in Figure 4.10(a), for only 10 minutes of data collected from smartphones, about half of the floor is covered by the mapped passive fingerprints. As the opportunistic RSS trace mapping goes on, at 30 minutes, almost every location on the floor is covered by at least one passive fingerprint as shown in Figure 4.10(c).

Since the passive fingerprints characterize the RSS property of each physical location on the floor, the RSS distributions can be directly visualized after the fingerprints are mapped. Figure 4.10(d), (e), and (f) show the RSS distribution evolution of WiFi monitor 1 at time 10 minutes, 15 minutes and 30 minutes. The darker color represents higher RSS values. We can see that with more trace mappings being performed, the RSS distribution of the whole floor becomes more and more complete from 10 to 30 minutes. In Figure 4.10(f), the RSS distribution of WiFi monitor 1 is almost completed for the whole floor at 30 minutes. The reconstructed RSS distribution from the trace mappings shows that the highest RSS values are observed when the smartphones are near Monitor 1, and that the RSS measurements become smaller when the smartphones are farther away, which is consistent with the theoretical wireless signal propagations. Unlike the

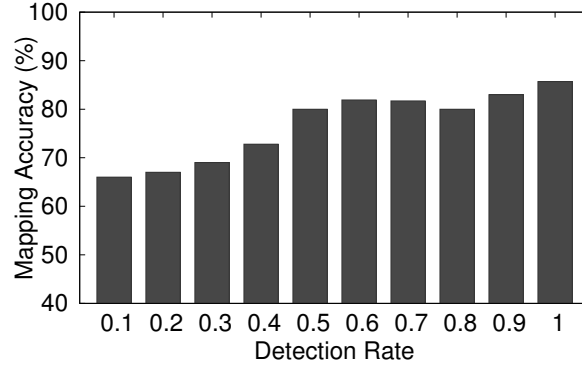


Figure 4.13: Impact of Sparsity of Detection

model-based RSS estimation, the reconstructed RSS distribution unveils the real RSS property at each mapped location, which helps to provide the fine-grained localization performance. The RSS distributions for all WiFi monitors are represented by the mapped passive fingerprints that are stored in the constructed database, which are then used to achieve real-time localization in the localization phase.

4.3.2.3 Impact of Sparsity of Transmission Detections

One concern about trace mapping is the density of transmissions that can be detected from WiFi monitors when smartphone users are moving indoors. Intuitively, denser detections provide more information to the signal evolution patterns between two landmarks, and will help to infer the correct route in-between. However, if there are very few detections, the system might not be able to infer the correct walking routes. To understand the impact of the sparsity of transmission detections on the trace mapping, we analyze the mapping accuracy with different level of transmission sparsity.

On average the transmission detection rate for the original RSS traces is about one detection per second. We vary the detection rate by randomly dropping detections from the traces with different probabilities. As shown in Figure 4.13, a 0.5 detection rate is approximated by dropping each detection from the trace with 50% probability, resulting an average detection rate of about 0.5 detections per second. We can see that the transmission detection rate does affect the mapping performance, and the mapping performance increases as the detection rate increases. However, even with the 0.5 detection rate, the mapping

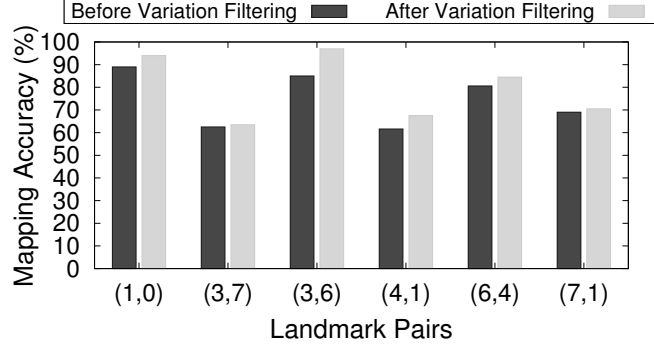


Figure 4.14: Trace Mapping For Traces with Speed Variations

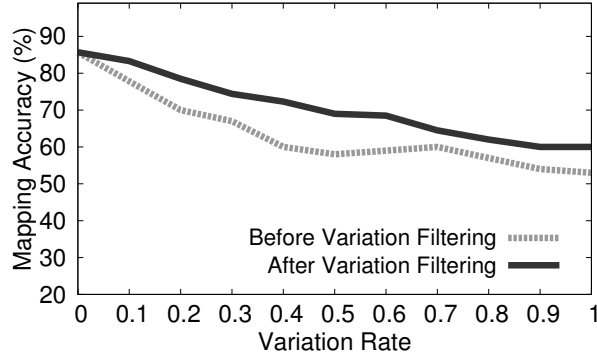


Figure 4.15: Performance of Variation Filtering

accuracy remains as high as 80%. In the worst case with a 0.1 detection rate, the mapping accuracy is around 66%. The results show that the mapping performance remains high even when the detection rate significantly is reduced, and that RSS trace mapping can therefore be performed in dynamic environments when smartphone detections are sparse.

4.3.2.4 Impact of Variations in the Walking Speed

To understand the impact of variations in the walking speed, we use the traces annotated with speed variations of users to perform trace mapping. As the walking route inference assumes a stable walking speed, the speed variation degrades the final mapping performance. As shown in Figure 4.14, the mapping accuracy decreases for all landmark pairs, compared with the traces without speed variation shown in Figure 4.11. The average mapping accuracy drops to 53.1% for all landmark pairs. The results indicate that the speed variation filtering is necessary to avoid introducing large false mappings.

SpiLoc filters traces with speed variation by looking at the RSS divergence in the traces. As shown in Figure 4.14, variation filtering using RSS divergence

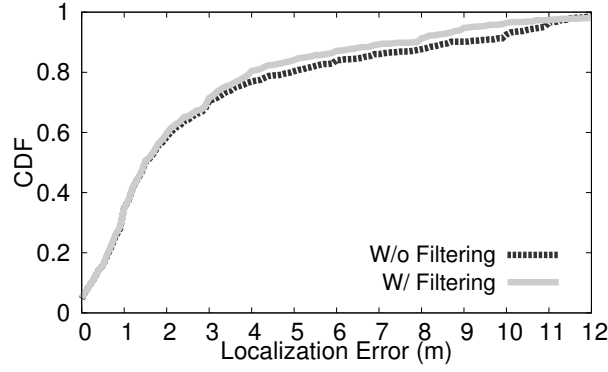


Figure 4.16: Localization Performance

improves the mapping performances for all landmark pairs. Figure 4.15 shows the mapping performance in the presence of different fraction of variation traces. As the fraction of traces with variation increases, the mapping accuracy drops from 85% to 53.1% without variation filtering. The variation filtering successfully improves the mapping accuracy by about 10% for all cases. If 20% of the collected traces have speed variation, the final mapping accuracy is still close to 80% after filtering. Even if the variation fraction raises to 50%, the final mapping accuracy remains as high as 70% after variation filtering. The results show that variation filtering in SpiLoc is important to keep the trace mapping robust in practice.

4.3.2.5 Localization Performance

We evaluate the final localization using the constructed passive fingerprint database. For each ground truth location entered by the user, we extract the RSS readings with the same timestamp from all WiFi monitor traces as input and calculate the location using Equation (4.5). The error is obtained by comparing the Euclidean distances between the ground truth locations and the estimated locations.

Figure 4.16 shows the CDF of the localization error. Without noise filtering for the bootstrapped passive fingerprint database, localization in SpiLoc achieves a 2.94m localization error on average. More than 70% of the errors are within 3 meters. Compared with the model-based passive localization scheme used in [56], SpiLoc leverages the bootstrapped fingerprint database and achieves a much more fine-grained localization result. As false trace mappings introduce noise to the fingerprint database, the noise filtering further improves the localization ac-

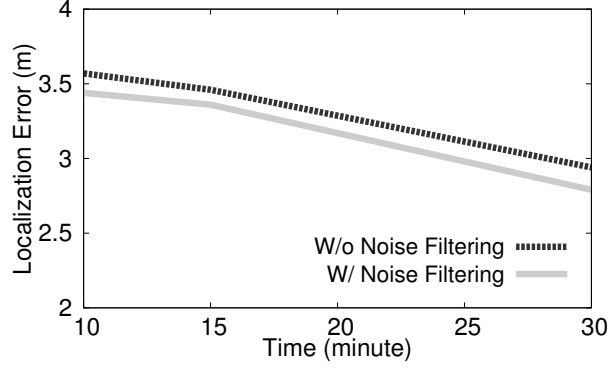


Figure 4.17: Localization Error with Different Input Data

System	Category	Localization Error	Effort
RADAR [13]	Active	2~5m	Time consuming site survey
Horus [89]	Active	1~2m	Time consuming site survey
Zee [63]	Active	1~ 3m	Require accurate floor plan
PiLoc	Active	1~ 3m	Dynamically bootstrapped
WiFi Tracking [56]	Passive	~70m	Coarse-grained passive localization
SpiLoc	Passive	2~3m	Dynamically bootstrapped

Table 4.2: Comparison with Different Localization Schemes

curacy. With noise filtering using Mahalanobis distance based outlier detection, the final localization error is reduced to around 2.7m. As shown in Figure 4.16, large errors introduced by the fingerprint noise are reduced after noise filtering.

As SpiLoc exploits a crowdsourcing scheme to opportunistically bootstrap the localization database, the number of input RSS traces also has an impact on the final localization accuracy. Figure 4.17 shows the localization error with different input trace sizes. With only 10 minutes of RSS traces from all WiFi monitors, the system achieves a 3.5m localization error. After 30 minutes of signal traces are collected, the localization error is gradually reduced to around 2.8 meters after noise filtering. Note that the training speed would have become greater as more smartphones contribute data in the bootstrapping phrase. As more RSS traces are collected to update the fingerprint database, the system will gradually adapt to the environmental changes, and provide stable localization performance over time. The final localization error for all testing data settles 2.76 meters.

Table 4.2 summarizes the differences between SpiLoc and other localization schemes. Unlike active localization schemes, SpiLoc as proposed in this work requires no active cooperation of the smartphones to infer their locations. Although SpiLoc exploits the opportunistic scheme to automatically bootstraps it-

self for passive localization, it achieves performance comparable to that of active localization schemes, which either require time-consuming site surveys or heavily rely on the cooperation of smartphones. The passive tracking for smartphones proposed in [56] aims at achieving coarse-grained tracking for smartphone. SpiLoc, on the other hand, achieves a fine-grained localization performance while requiring no additional costs.

4.4 Discussion

4.4.1 Dedicated Site Surveys

As with the site-survey process in active localization, it is possible to perform dedicated site surveys to construct the passive fingerprint database manually. However, such dedicated site surveys are labor-intensive and time-consuming, which makes the system hard to deploy on a large scale. In addition, the site-survey approach builds static fingerprint databases, which are vulnerable to environmental changes. In SpiLoc, on the other hand, we exploit the opportunistic trace mapping approach to automatically build and update the fingerprint database, which significantly reduces the start-up costs and maintenance efforts, making the system scalable and adaptive.

4.4.2 Prompting Extra Transmissions

As the performance of trace mapping is affected by the sparsity of transmission detections, the density of transmission detections has an assignable impact on the quality of the constructed fingerprint database and the final localization accuracy. Although the route inference approach proposed in SpiLoc works with sparse transmission detections, maximizing the number of detections is important to further improve the performance of the system.

Several techniques have been proposed in the literature. For example, one useful technique proposed in [56] is to let the WiFi monitors emulate popular SSIDs as the smartphones will automatically connect to these popular WiFi hotspots. For example, when popular SSIDs such as “attwif” or “tmobile” are advertised in the U.S., phones will likely send association requests [56]. Other useful techniques such as sending RTS to trigger CTS responses [56] will also

increase the detection chances.

All these techniques can be seamlessly integrated into SpiLoc as an additional component to prompt more transmissions from smartphones to increase the mapping performance and final localization accuracy.

4.4.3 Open Area

The RSS trace mapping in SpiLoc works in the office environments in which walking routes connecting landmarks follow indoor walking paths. In indoor open areas, however, users have no walking paths to follow, and it becomes less feasible to infer the routes users travel purely based on the RSS measurements. This is one limitation of SpiLoc. However, if the WiFi monitors are dense enough and we can determine the straight walking routes connecting landmarks directly, we can map the whole floor even in open spaces. One possible way to detect straight walking routes is using strict temporal filters to filter out non-direct routes. We leave this open problem as a possible subject of future research.

4.4.4 Privacy Risks

As the passive localization scheme requires no active participation of smartphones, smartphone locations may be unintentionally revealed to third parties. Since the applications of passive tracking, such as passive counting or customer flow analysis, usually do not require specific MAC addresses, one simple but effective approach to avoid privacy risks is to anonymize or replace the smartphones' MAC addresses at the server side. Although there might be more sophisticated approaches to address the privacy problem, these are beyond the scope of this work.

4.5 Summary

In this chapter, we propose SpiLoc, a passive localization system that automatically detects landmarks that appear in the RSS traces captured by WiFi monitors, and infers the most likely walking routes that connect these landmarks. By mapping the traces collected between landmarks, SpiLoc bootstraps the passive fingerprint database for localization. As the fingerprints alleviate

the multi-path problem and characterize the RSS property of each indoor location, SpiLoc can achieve fine-grained localization with a mean error of 2.76 meters. Since SpiLoc requires no dedicated calibration and adaptively updates itself every time an RSS trace mapping is performed, it can be easily deployed to dynamic environments for fine-grained passive localization.

Chapter 5

A²Loc: Accuracy Awareness of Wireless Indoor Localization

5.1 Introduction

State-of-the-art research on fingerprint-based indoor localization focuses on either improving the accuracy of the location estimation [88, 49, 75], or reducing the time and effort taken to construct the fingerprint database [86, 76, 74]. Participatory sensing based indoor localization systems such as PiLoc and SpiLoc automatically generate radio maps for localization. However, as there is no efficient way to assess the quality of the output radio maps, it is hard to get direct feedback about the performance of the system. An efficient approach to estimate the localization accuracy based on radio maps will therefore be very useful in understanding the performance of any fingerprint-based indoor localization system. In view of this, the major objectives of our work described in this section are (1) designing an efficient approach to get direct fine-grained localization accuracy estimation using only the constructed radiomaps as input; (2) developing an approach to extract useful information, such as localization landmarks that exist in the system; (3) providing guidelines for localization algorithm selection and parameter tuning, such as the subset selection of WiFi access points in practice.

The main idea of this work is as follows: given a set of radio signal fingerprints collected, a Gaussian process (GP) [66] approach is used to model the signal distribution of access points that covers the area of interest. Using the signal distribution model derived, random sampling is performed to simulate

the collection of fingerprint values collected at each location of interest during localization. Given a particular localization algorithm, the mapped location in the system can be determined. The average localization error of each location in the area of interest can then be estimated even though the original set of data collected as input may not have been sufficient for localization purposes on its own.

By decoupling radio map construction and localization, and with the ability to estimate the accuracy of the localization system over the area of interest, our system can achieve the following: (1) It is now possible to systematically compare different localization algorithms under different environmental settings. (2) Landmarks, or locations with high localization confidence, can be easily identified and used to further improve the accuracy. (3) The set of APs that can provide better accuracy for the entire area of interest can be identified, as opposed to using all APs available or a set of APs that may be good locally but not for the entire area.

Though several systems have been proposed in the literature that deal with wireless signal modeling and fingerprint-based localization accuracy analysis [26, 84, 91, 38, 71, 15, 30], all of them focus on improving the performance of localization algorithms by modeling signal properties [26, 84, 91, 71, 30], designing optimal AP placements [15], or modeling localization uncertainties [38]. To the best of our knowledge, this work is the first systematic study to provides a direct quality assessment of radio maps and give fine-grained performance estimations to fingerprint-based localization systems. We believe that it has the potential to be integrated into future fingerprint-based localization systems to provide direct feedback about the accuracy levels of the system in use, and useful guidelines to achieve better accuracy.

To validate our approach, we evaluate the system in two different indoor environments covering more than 300 m^2 . In both environments, point-level, region-level and floor-level error estimation are evaluated with three different localization metrics and more than 20,000 testing data points. For point level accuracy, the evaluation results show that the difference between GP estimation and ground truth is small, demonstrating that accuracy awareness provides an accurate and practical method of assessing fingerprint-based localization sys-

tems. In addition, we are able to successfully identify five landmarks with high localization confidence in the area localized, and find the minimum AP subsets that should have been selected to achieve better accuracy.

5.2 Accuracy Awareness

5.2.1 Preliminaries

The RSS of the wireless access point at each location has been characterized in the literature as a Gaussian distribution [28, 88, 33, 26]. On the other hand, to model the signal strength propagation continuously over the whole field, Gaussian process is used to capture the spatial correlation that exists in signal strength distribution [27, 26, 84]. A Gaussian Process (GP) [66] is a Bayesian non-parametric model that performs non-linear regression on the training data $\mathcal{D} = \{(\mathbf{x}_i, y_i) | i = 1, \dots, n\}$ to estimate the distribution over functions f that generate the data. That is,

$$y_i = f(\mathbf{x}_i) + \varepsilon \quad (5.1)$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is a d dimensional input value, y_i is the observation value, and ε is a zero-mean noise term with known covariance σ_n^2 . Gaussian processes allow spatial correlation between measurements and are fully specified by GP priors. Therefore, function $f \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ is a GP with mean function $\mu(\mathbf{x})$ and covariance function, or kernel, $k(\mathbf{x}, \mathbf{x}')$, where:

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] \quad (5.2)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))] \quad (5.3)$$

The choices of the kernel function characterize the property of GPs, and the most widely used kernel is the *squared exponential* function [26]:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2l^2} |\mathbf{x} - \mathbf{x}'|^2) \quad (5.4)$$

where σ_f^2 is the variance of observation value and l is the length scale that decides how strongly the correlation between different points drops off [26]. Assuming additive independent identically distributed Gaussian noise ε and noise covari-

ance σ_n^2 [66], the covariance between observations becomes:

$$\text{cov}(f(\mathbf{x}), f(\mathbf{x}')) = k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \delta_{\mathbf{x}, \mathbf{x}'} \quad (5.5)$$

Here $\delta_{\mathbf{x}, \mathbf{x}'} = 1$ if \mathbf{x} and \mathbf{x}' are the same point, and 0 otherwise. After the prior is specified, the Gaussian process posterior is obtained from the training data \mathcal{D} . Therefore with GP priors and training data, prediction of the unobserved function value at any arbitrary location \mathbf{x}^* can be made [84]:

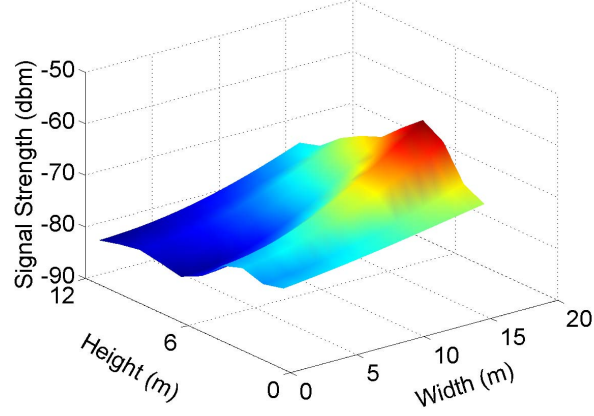
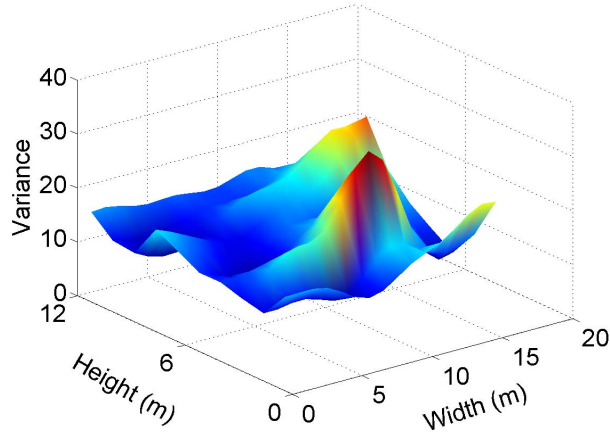
$$\mu_{\mathbf{x}^*|\mathcal{D}} = \mu_{\mathbf{x}^*} + \Sigma_{\mathbf{x}^*\mathcal{D}} \Sigma_{\mathcal{D}\mathcal{D}}^{-1} (y_{\mathcal{D}} - \mu_{\mathcal{D}}) \quad (5.6)$$

Here $\mu_{\mathbf{x}^*}$, $\mu_{\mathcal{D}}$ are the mean values of the data points and are specified by the GP prior $\mu(\mathbf{x})$. $\Sigma_{\mathbf{x}^*\mathcal{D}}$ is the $1 \times n$ vector of covariance between x^* and the n training data \mathcal{D} , and $\Sigma_{\mathcal{D}\mathcal{D}}^{-1}$ is the $n \times n$ covariance matrix of the training data. Both $\Sigma_{\mathbf{x}^*\mathcal{D}}$ and $\Sigma_{\mathcal{D}\mathcal{D}}^{-1}$ are calculated using Equation (5.5). With this formulation, the observation value at any arbitrary location in the field can be predicted conditionally on the training data.

To model the signal strength distribution of the access points covering a certain area, input $\mathbf{x} = (x_h, x_v)$ is a two dimensional vector specifying the horizontal and vertical coordinates of the location. The observation value y_i is the signal strength received at the given location. Note that the input data \mathcal{D} here can be obtained from the fingerprint database, or radio map, which is generally required and constructed by any fingerprint-based localization systems in the offline calibration phase in order to perform localization.

The radio map contains a sequence of records $(\mathbf{x}, \mathbf{fp})$, which associates wireless fingerprints \mathbf{fp} to each location \mathbf{x} . Each fingerprint $\mathbf{fp} = (BSSID_i, r_i | i = 1, \dots, k)$ consists of signal strength readings r of all k WiFi BSSIDs (MAC addresses of access points) observable. Hence for each BSSID in the system, the training data $\mathcal{D} = \{(\mathbf{x}_i, r_i) | i = 1, \dots, n\}$ is available. With the availability of the training data, Gaussian processes can be applied to characterize the signal strength distribution of the whole area.

The squared exponential kernel in Equation (5.4) assumes the same length scale in all input dimensions. However, in practice the effect of horizontal or vertical dimensions to signal strength can be different due to the physical set-


 Figure 5.1: Mean Prediction ($\mu_{\mathbf{x}^*|\mathcal{D}}$)

 Figure 5.2: Variance Prediction ($\sigma_{\mathbf{x}^*|\mathcal{D}}^2$)

tings. For example, there could be a wall in the horizontal dimension, resulting in the fast decay of signal strength in only this dimension. To model this effect, we use separate length scales l_h and l_v in each dimension on modeling the signal strength:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left[-\frac{1}{2}\left(\frac{(x_h - x'_h)^2}{l_h^2} + \frac{(x_v - x'_v)^2}{l_v^2}\right)\right] \quad (5.7)$$

The mean function and covariance function characterize the signal strength model. To handle the mean shift problem, we set mean function $\mu(\mathbf{x}) = -100$, so that those locations that are not able to receive any signal strength from certain access points will converge to mean -100dbm in its model. The covariance function contains four parameters $\theta = \langle \sigma_n, \sigma_f, l_h, l_v \rangle$.

One advantage of the GP is that it is a non-parametric model, and therefore no parameters need to be specified beforehand, all parameters are learned from the training data by maximizing the log likelihood using the conjugate gradient

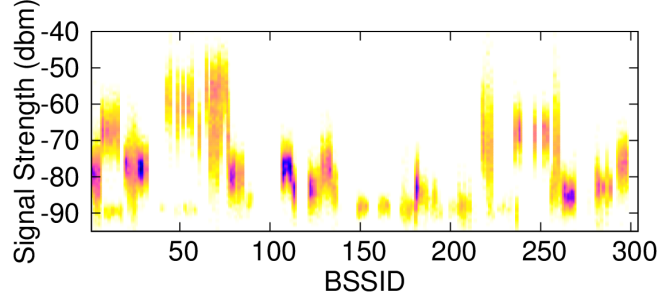


Figure 5.3: Gaussian Process Sampling

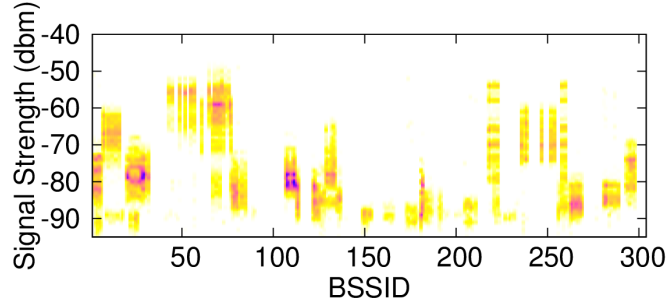


Figure 5.4: Ground Truth Phone Sampling

decent algorithm [27]. Figure 5.1 shows the GP estimation of the mean signal strength value for one access point covering a $20 \times 12m^2$ indoor area. Note that even though the GP also provides uncertainty measurement for Equation (5.6) (e.g., the variance of the predicted $\mu_{\mathbf{x}^*|\mathcal{D}}$ [27]), it only measures the “spatial uncertainty” of the predicted mean. This uncertainty is different from the “temporal uncertainty”, which is the variance of signal strength at each location at various times. The temporal uncertainty provides the likelihood measurement for the signal strength. To model temporal uncertainty, we treat variance as the second variable and train a second GP for the same access point, using mean function $\mu(\mathbf{x}) = 0$ and the same covariance function (5.7) from input data \mathcal{D} . Figure 5.2 shows the RSS variance estimate $\sigma_{\mathbf{x}^*|\mathcal{D}}^2$ at each location \mathbf{x}^* for the access point. With $\mu_{\mathbf{x}^*|\mathcal{D}}$ and $\sigma_{\mathbf{x}^*|\mathcal{D}}^2$, we are now able to obtain the likelihood of each signal strength value at an arbitrary location for each access point.

5.2.2 Accuracy Awareness

In this section, we study the accuracy awareness of fingerprint-based localization systems and applications enabled in three different granularities based on the GP signal strength model.

5.2.2.1 Point-level Accuracy

Point-level accuracy is commonly used in most localization systems to measure performance. Such accuracy depends on the ability of fingerprints to uniquely identify a particular location. Hence, the fingerprints at different locations should display sufficient *location diversity*. In our work, signal strength models of access points derived from the training data \mathcal{D} (radio map) provide the mean signal strength value $\mu_{\mathbf{x}^*|\mathcal{D}}$ and variance $\sigma_{\mathbf{x}^*|\mathcal{D}}^2$ of each access point at each location \mathbf{x}^* . We can then use this information to get the likelihood estimate for fingerprints and simulate fingerprint sampling at each location during localization to get the error estimate.

(1) Error Estimation

Errors in fingerprint-based localization come from the fact that sampled signal strengths of access points fluctuate and can be different from the fingerprints in the radio map. By chance, they will be mapped to different locations. To characterize the average localization error $E(\mathbf{x})$ at one location $\mathbf{x} = (x_h, x_v)$:

Algorithm 3: Fingerprint Sampling Algorithm

```

1 Input: Location  $\mathbf{x}$ , mean  $\mu_{\mathbf{x}|\mathcal{D}}$ , variance  $\sigma_{\mathbf{x}|\mathcal{D}}^2$ ,  $k$ 
2 Output: Sampled fingerprint fp
3 for  $i = 1:k$  do
4   | If  $r_i$  hasn't been assigned, with probability  $p_i$  set  $r_i = \text{rand}(\mu_{\mathbf{x}|\mathcal{D}}, \sigma_{\mathbf{x}|\mathcal{D}}^2)$ ,
   |   otherwise set  $r_i = -100$ ;
5   | For all  $j > i$ , set  $r_j = r_i$  if  $S_{ij} < \tau$ ;
6 end
```

$$E(\mathbf{x}) = \sum_{\mathbf{x}'} p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x}) \cdot d(\mathbf{x}, \mathbf{x}') \quad (5.8)$$

where \mathbf{x}' is the reported location by the localization algorithm \mathcal{L} , and $p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x})$ is the probability that the localization algorithm \mathcal{L} reports \mathbf{x}' when users are actually in \mathbf{x} , and $d(\mathbf{x}, \mathbf{x}')$ is the Euclidean distance between two locations on the 2D plane in meters.

For evaluation purposes, the area of interest is discretized into a number of locations. By taking the average of all possible locations, the expected error of each location can be obtained. $p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x})$ is determined by the localization

algorithm \mathcal{L} and property of fingerprints \mathbf{fp} collected at these locations:

$$p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x}) = \sum_{\mathbf{fp}} p(\mathbf{fp}|\mathbf{x}) \cdot \delta_{fp} \quad (5.9)$$

Here $p(\mathbf{fp}|\mathbf{x})$ is the possibility that fingerprint \mathbf{fp} can be sampled at location \mathbf{x} . $\delta_{fp} = 1$ if $\mathcal{L}(\mathbf{fp}) = \mathbf{x}'$, that is, the localization algorithm maps \mathbf{fp} to location \mathbf{x}' and $\delta_{fp} = 0$ otherwise. The mapping of \mathcal{L} is deterministic once the fingerprint is given and the localization algorithm is chosen. The localization error $E(\mathbf{x})$ hence depends on the fingerprint characteristics and the algorithms used. To get the error estimate from (5.8) and (5.9), fingerprints need to be traversed. Although we already have the likelihood estimate for each fingerprint using Gaussian processes, consider a floor with k access points, with each access point having q different signal strength readings, we have q^k different fingerprints.

(2) Fingerprint Sampling

It is not feasible to traverse the fingerprint space in practice when k can easily exceeds 100 and $q = 71$ when signal strength ranges from $[-100, -30]$. Instead, we use Monte Carlo sampling approach [34] to simulate fingerprint based localization and get the error estimate for each location.

To model the real fingerprint readings, each access point is generally considered to be independent [33, 27, 26]. This assumption is made based on the fact that access points are physically separate. However, modern access points allow multiple BSSID beacon settings, which make the access points able to broadcast multiple BSSID addresses [3]. Therefore, different signal strength readings of different BSSIDs can belong to the same access point, resulting in the readings of these BSSIDs to be mostly identical. These duplicated BSSIDs are recorded by the sampling devices such as smartphones in the radio map during the calibration phase and are used to perform localization during the online phase as long as they can be received at the location. It is therefore not correct to assume independence between these BSSIDs. We use the following metric to detect these duplicated BSSIDs:

$$S_{ij} = \frac{\sum_{i=1}^m |r_i - r_j|}{m \cdot |r_{min}|} \quad (5.10)$$

where m is the total number of fingerprints collected in \mathcal{D} , and r_i and r_j are the signal strength of two BSSIDs, and are set to -100 if the BSSIDs are not detected in this fingerprint. $r_{min} = -100$ is the minimum signal strength observable. S_{ij} should be small if these two BSSIDs are broadcast by the same access point. In the sampling, BSSIDs with S_{ij} less than the threshold τ are set to have the same signal strength. For this work τ is set to 0.005. At each location, the probability p_i that one BSSID can be received can also be learned from the training data \mathcal{D} . As shown in Algorithm 3, for a fingerprint \mathbf{fp} containing k BSSIDs, the signal strength r of each BSSID is sampled randomly from the mean $\mu_{\mathbf{x}^*|\mathcal{D}}$ and variance $\sigma_{\mathbf{x}^*|\mathcal{D}}^2$ learned from the Gaussian processes with probability p_i . Otherwise it is set to -100 , indicating that the BSSID is not observed in this fingerprint.

Figure 5.3 shows the fingerprints sampled by the sampling algorithm for 304 BSSIDs at one floor in one randomly selected location. Figure 5.4 shows the ground truth fingerprints sampled by smartphones at the same location. We can see that the GP-based sampling algorithm follows actual fingerprint samples fairly well and provides a “smoother” distribution. With the sampling algorithm, we are now able to simulate fingerprinting at arbitrary locations on this floor to get the error estimate.

(3) Sample Size Determination

Each fingerprint \mathbf{fp} sampled by Algorithm 3 provides one error estimate $e(\mathbf{x})$ for the location \mathbf{x} :

$$e(\mathbf{x}) = d(\mathbf{x}, \mathcal{L}(\mathbf{fp})) \quad (5.11)$$

To estimate the average error $E(\mathbf{x})$ with random sampling, we need to decide the minimum sample size n_e to achieve confidence interval α . From statistical theories [24]:

$$\frac{\overline{e(\mathbf{x})} - E(\mathbf{x})}{S/\sqrt{n_e}} \sim t(n_e - 1) \quad (5.12)$$

where $\overline{e(\mathbf{x})}$ is the mean of all n_e estimates of $e(\mathbf{x})$, S is the standard derivation of the n_e samples, and $t(n_e - 1)$ is the t-distribution with $(n_e - 1)$ degrees of freedom [24]. The confidence interval $[-\frac{S}{\sqrt{n_e}} \cdot t_{\alpha/2}(n_e - 1), \frac{S}{\sqrt{n_e}} \cdot t_{\alpha/2}(n_e - 1)]$ ensures the error estimation with α confidence. We set $\alpha = 99\%$ here.

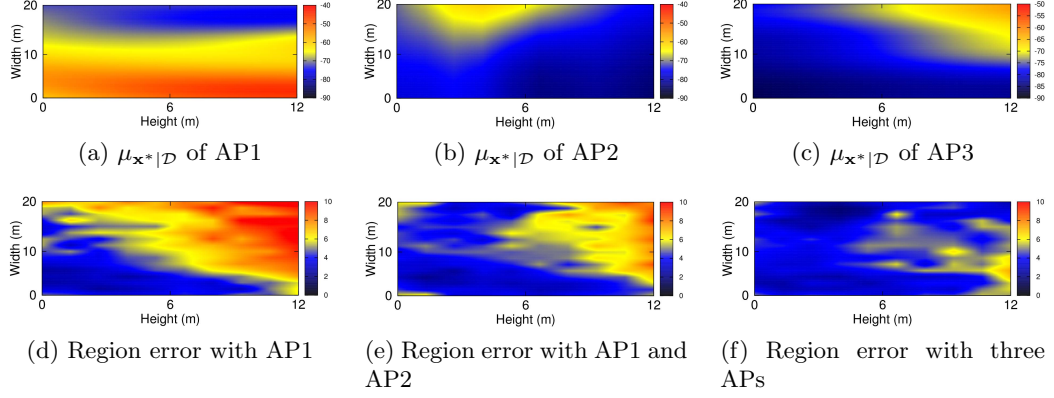


Figure 5.5: Region Error Evolution

To make the average error estimate less than ϵ :

$$2 \frac{S}{\sqrt{n_e}} \cdot t_{\alpha/2}(n_e - 1) < \epsilon \quad (5.13)$$

with this, the minimal sample size n_e can be calculated. ϵ is the maximum estimation error and is set to 0.1m. Algorithm 4 is the final algorithm for the average localization error at each location \mathbf{x} . n_0 is the initial sample size and is set to 100. After that the sample size keeps increasing until it meets the constraint set by Equation (5.13). The average error $\mathbf{E}(\mathbf{x})$ is then obtained from the sampling algorithm.

Algorithm 4: Error Estimation Algorithm

```

1 Input: Location  $\mathbf{x}$ , localization algorithm  $\mathcal{L}$ 
2 Output: Average localization error  $\mathbf{E}(\mathbf{x})$ 
3 while Size of  $e(\mathbf{x})$  list  $< n_0$  or Equation (5.13) not met do
4   | Sample another fp using Algorithm 3;
5   | Calculate  $e(\mathbf{x})$  using (5.11);
6   | Add  $e(\mathbf{x})$  to the sampled error list;
7 end
8 Return the mean of all sampled  $e(\mathbf{x})$  as  $\mathbf{E}(\mathbf{x})$ ;
    
```

(4) Landmark Detection

While the error estimation algorithm provides a new way to analyze the error characteristics of all locations, it also provides opportunities to extract other useful information to enhance the performance of conventional fingerprint-based localization systems. The concept of landmark is widely used in localization systems. However, how to identify landmarks automatically is less discussed. A

landmark is a place in which, once a user is localized to it, the system should have high confidence that the user is indeed there. Landmarks are widely exploited in various localization systems to improve their performance. For example, landmarks are used to reset the dead reckoning error [76], or simply to increase the localization accuracy [20]. With the ability to estimate the point-level localization error, we are also able to detect landmarks that are present in the system.

At each given location \mathbf{x} , the confidence that the user is actually in \mathbf{x} when the mapped location from the localization algorithm \mathcal{L} is \mathbf{x}' is $p_{\mathcal{L}}(\mathbf{x}|\mathbf{x}')$. $p_{\mathcal{L}}(\mathbf{x}|\mathbf{x}')$ can be obtained using Bayes' theorem:

$$p_{\mathcal{L}}(\mathbf{x}|\mathbf{x}') = \frac{p_{\mathcal{L}}(\mathbf{x}) \cdot p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x})}{p_{\mathcal{L}}(\mathbf{x}')} \quad (5.14)$$

Here $p_{\mathcal{L}}(\mathbf{x})$ is the probability that the user is in location \mathbf{x} of the indoor environment. We assume all locations are equally likely, namely that, the probability of the user being in each location is the same. Then $p_{\mathcal{L}}(\mathbf{x}) = 1/n_l$, where n_l is the total number of discrete locations on the radio map. For example, if we have 100 candidate locations on the radio map, $p_{\mathcal{L}}(\mathbf{x}) = 0.01$ for each location \mathbf{x} . $p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x})$ is the probability that the reported location by the localization algorithm is \mathbf{x}' when the user is actually in \mathbf{x} . After running the point-level error-estimation algorithm, we are able to obtain the number of samples $n_{\mathbf{x}'}$ that fall into location \mathbf{x}' out of the total number of samples n_e in location \mathbf{x} . Also, $p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x}) = n_{\mathbf{x}'}/n_e$. $p_{\mathcal{L}}(\mathbf{x}')$ is the probability that the mapped location is \mathbf{x}' when fingerprint-based localization is performed on this indoor floor. Similarly, $p_{\mathcal{L}}(\mathbf{x}')$ can also be easily calculated from the result of the error-estimation algorithm, in the same way as $p_{\mathcal{L}}(\mathbf{x}'|\mathbf{x})$.

In this way, the confidence $p_{\mathcal{L}}(\mathbf{x}|\mathbf{x}')$ is obtained for all the possibilities of location \mathbf{x} . We are especially interested in finding the confidence when \mathbf{x} and \mathbf{x}' are the same location. If the confidence is high enough, the location becomes a landmark, and can be further exploited by localization systems. We provide our evaluation of landmark detection in Section 5.3.

5.2.2.2 Region-level Accuracy

While the point-level accuracy provides the error characteristics of each location in an indoor environment, viewing it at a coarse granularity gives a different

perspective of the system behavior. In this section, we analyze the error characteristics at the region-level.

A region here consists of those nearby locations with similar localization errors. The region-level error summarizes the region error distribution and can help to identify blind spots for the localization system. By identifying these regions, we have opportunities to improve these regions accordingly. For example, one possible way to improve the poor region performance in fingerprint-based indoor localization system is to place another access point in this region. Placing additional access points will increase the “uniqueness” of the fingerprints in this region and hence reduce the localization error for the whole region. Figure 5.5 illustrates the idea and shows the region error evolution when more access points are added.

For this figure, an indoor $20 \times 12m^2$ indoor environment is measured. Three access points located at three different indoor locations are added one by one to the large error region. Figure 5.5(a), Figure 5.5(b), and Figure 5.5(c) show the mean RSS value distribution of these three access points, which reflect their relative locations on this floor. For example, AP1 is located at the bottom and AP2 at the top left corner. Errors are obtained from Algorithm 4.

After AP1 is added to the system, regions with errors of less than four meters and greater than four meters are identified, and are shown in Figure 5.5(d). To improve the region with larger errors, we place another AP2 into the system and the result is shown in Figure 5.5(e). We can see that parts of the regions with larger errors are successfully converted to regions with errors of less than four meters and region with errors greater than 8 meters are eliminated with only two access points. Adding AP3 to the poor regions further improves the performance, and converts some of those poorer regions into regions with smaller errors.

With error-estimation algorithm and region-level analysis, the error distribution of the indoor floor is visualized and the impact of each access point on the whole system also becomes easily observable. This capability is useful in identifying poor performance regions, and deciding where to place new access points, or deciding which APs should be included in the fingerprint database.

5.2.2.3 Floor-level Accuracy

The overall performance of a localization system depends on many factors, such as the localization algorithm \mathcal{L} used, and the deployment of access points. The average error of the whole floor E_f is an important metric that is widely used in the literature to characterize the localization performance. Here,

$$E_f = \sum_{i=1}^{n_l} E(\mathbf{x}_i) / n_l \quad (5.15)$$

is the average point-level error of all n_l locations on the same floor. We focus on the floor-level accuracy and study the factors that affect the overall accuracy in this section.

(1) Localization Algorithm Selection

As discussed in Section 5.2.2.1, a fingerprint-based localization algorithm \mathcal{L} maps a fingerprint \mathbf{fp} to a location \mathbf{x}' . Many localization algorithms have been proposed in the past decades [12, 88, 52], these algorithms have different reported accuracies and might be suitable for various environmental settings. However, there is no efficient way to compare them and choose which algorithm to use in a given environment to get the best accuracy. For example, simple nearest neighbor mapping algorithm [12] (NN1) is widely used due to its simplicity. However, in certain environments the top-3 nearest neighbor mapping (NN3) or top-5 nearest neighbor mapping (NN5) might give better accuracy. The accuracy of mapping depends on both the fingerprint characteristics and the mapping algorithm. While the GP-based sampling and estimation algorithm discussed in the previous section provides an error estimate, the error is also largely dependent on the localization algorithm \mathcal{L} . The different floor-level accuracies E_f obtained by varying \mathcal{L} provide a direct comparison of localization algorithms. This capability provides guidance for choosing the most suitable localization algorithms. We provide our evaluation of algorithm selection in Section 5.3.

(2) Subset Selection

The advantage of wireless fingerprint-based localization is that it leverages existing wireless infrastructures. With regard to the received signal strength of

all access points, a natural question to ask is whether it is optimal to use all RSS values collected for localization, or to use only a selected subset of them. What is the best possible accuracy we can achieve with these already deployed access points? Also, in certain cases, users might want to reduce the size of the fingerprints to reduce networking costs or storage costs by including only a subset of BSSIDs into the fingerprints. What is the minimum number of BSSIDs we can use to achieve a certain accuracy? To the best of our knowledge, no existing works in the literature provide answers to these questions. In this section, we therefore discuss the BSSID subset selection problem to address all the above questions.

These questions can be answered if the following optimization problem can be solved:

$$\begin{cases} \text{minimize} & E_f \\ \text{subject to} & ||\mathbf{S}_B|| \leq k \end{cases} \quad (5.16)$$

where \mathbf{S}_B is the subset of BSSIDs selected from all BSSIDs to be used in the localization. k is the number constraint, which is usually the total number of BSSIDs we have in the indoor environment. The subset that minimizes the floor-level error E_f is the subset we should use in the localization, and the corresponding error is the minimum error we can achieve with all the deployed access points in the environment. We can also find out the minimum number of BSSIDs that can achieve a certain accuracy by increasing the value of k from 1. If the minimum error meets the requirements, it represents the minimum number of BSSIDs we need to use to achieve the required accuracy.

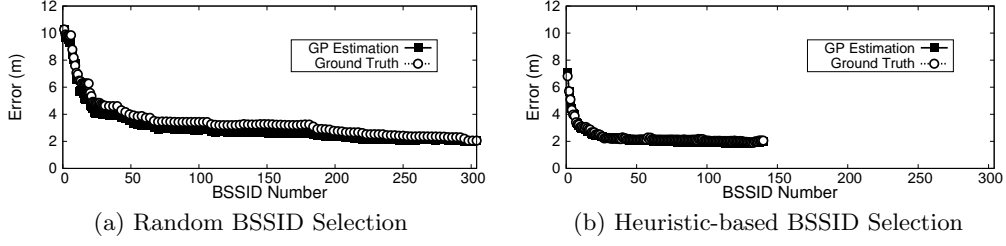
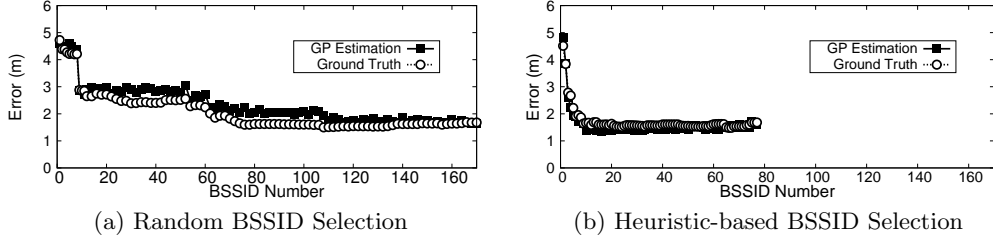
Algorithm 5: Subset Selection Algorithm

```

1 Input: Total BSSID set  $\mathbf{S}_{all}$ , number constrain  $k$ 
2 Output: Selected subset  $\mathbf{S}_B$ , Minimum floor-level error  $E_{min}$ 
3 Initialize  $\mathbf{S}_t$  to be an empty set.
4 while  $||\mathbf{S}_t|| < k$  do
5   | Add another not duplicated BSSID with most uniqueness to  $\mathbf{S}_t$ ;
6   | Calculate  $E_f$  using current subset  $\mathbf{S}_t$ ;
7   | if  $E_f < E_{min}$  then
8   |   |  $E_{min} = E_f$ ;
9   |   |  $\mathbf{S}_B = \mathbf{S}_t$ ;
10  | end
11 end

```

Once the subset \mathbf{S}_B is selected, the corresponding error E_f can be obtained

Figure 5.6: BSSID Selection ($240m^2$ Open Area)Figure 5.7: BSSID Selection ($72m^2$ Office Room)

easily with the error-estimation algorithm. For any BSSID subset, we can construct fingerprints \mathbf{fp} using Algorithm 3 with only the selected BSSIDs. Point-level error can be estimated using Algorithm 4 for all locations \mathbf{x} . Hence, the floor-level error E_f is also known. By comparing the floor-level error E_f , we can get the best BSSID subset with the minimum E_f . However, even though E_f can be obtained easily for each selected subset, solving the subset selection problem is NP-hard and it is impossible to enumerate all 2^k subsets when k can easily exceed 100. Instead, we use a heuristic-based method to approximate the optimal solution.

One heuristic that can be determined by the discussions in Section 5.2.2.2 is that adding access points to the poor performance region will increase the uniqueness of the fingerprints in the whole area, thereby reducing the overall error. Therefore, one can identify the poor regions and add access points with the most uniqueness to these regions. The uniqueness of $BSSID_i$ in these regions is defined by the range of average signal strengths specified by Equation (5.6). If the range is larger, more possibilities of fingerprint values are added to these regions, and hence the fingerprint diversity improves. Here, we set all regions with errors greater than 1.5 meters as regions to be improved.

Algorithm 5 illustrates the subset selection algorithm. Each time, an unselected BSSID that is not similar to any selected BSSIDs and the most uniqueness

is added until the number constraint is reached. The minimum error is stored in E_f and the selected subset in \mathbf{S}_B . The algorithm provides $\mathcal{O}(n)$ complexity and provides approximated solutions to the subset optimization problem.

5.3 Performance Evaluation of A^2Loc

The accuracy awareness based on the Gaussian process provides a direct assessment of different fingerprint-based localization systems. Two key concerns are how well the error estimation results fit the ground truth and how useful and the derived guideline information is. We discuss the evaluation results in this section.

5.3.1 Data

To evaluate the accuracy-awareness algorithms proposed, we collected data over a two-week period from a big $20 \times 12m^2$ indoor open area and a smaller $8 \times 9m^2$ office room. Three different phone models (Google Nexus 5, Samsung S3 and Samsung S4) were used to collect the WiFi radio map and the testing data. Each indoor environment was divided into $1 \times 1m^2$ grids and each grid was sampled for one minute to construct the radio map. The radio map was used as the training data \mathcal{D} to train the GP models for all access points. To collect the ground truth data, more than 20,000 phone fingerprint readings at random locations were collected as testing data to evaluate the performance.

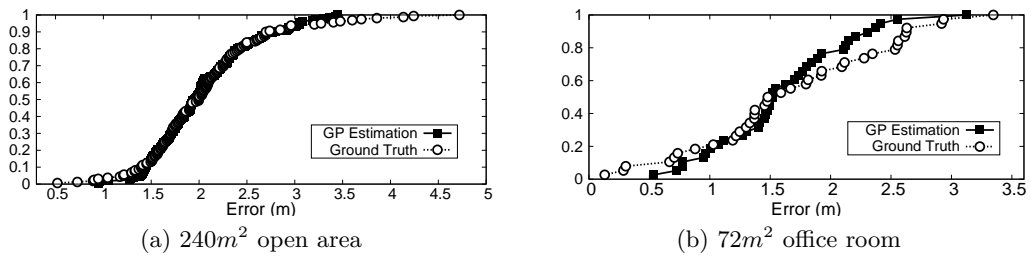


Figure 5.8: CDF of Point-level Error

5.3.2 Performance

5.3.2.1 Error Estimation

Figure 5.8 shows the CDF distribution of the point-level localization error in both indoor environments. The GP estimations are obtained with Algorithm 4 using the GP trained from the radio map. The ground truth error is measured using the testing data. In both cases, the localization algorithm \mathcal{L} is the same nearest neighbor matching (NN1). The CDF graphs in Figure 5.8 show the error characteristics of the indoor environment predicted by the GP-based error estimation algorithm and the ground truth. In both environments, the predicted CDF fit the ground truth error distribution very well, which means the predicted floor-level errors for both indoor environments are also very close to the ground truth. The GP-based estimation algorithm provide a smoother result, while the error distribution of the ground truth is more scattered, due to the noise in the fingerprints collected from the real phone readings. Figure 5.8 shows that the GP-based fingerprint sampling algorithm and the error estimation can successfully fit the error characteristics of the indoor environment and provide a close estimation of the localization error.

5.3.2.2 Landmark Detection

Landmark detection is a useful application enabled by the accuracy awareness. Locations with high localization confidence can be set to be landmarks to improve the system performance. Figure 5.9 shows the localization confidence distribution of all locations in the office room. The confidence is calculated using Equation 5.14. The threshold used for landmark detection is decided by different applications. If we set the threshold to be 0.7, five landmarks can be detected, as shown in Figure 5.9(a) (L1~L5). Once the localization algorithm has mapped the fingerprints to these locations, we should have high confidence that the mapping result is correct. Figure 5.9(b) shows the ground truth comparison for these five landmarks. The ground truth confidence is obtained using the testing data. The confidence of each location is the percentage of correct mappings when the localization algorithm maps the fingerprints to this location. The results show that the predicted confidence fit the ground truth well.

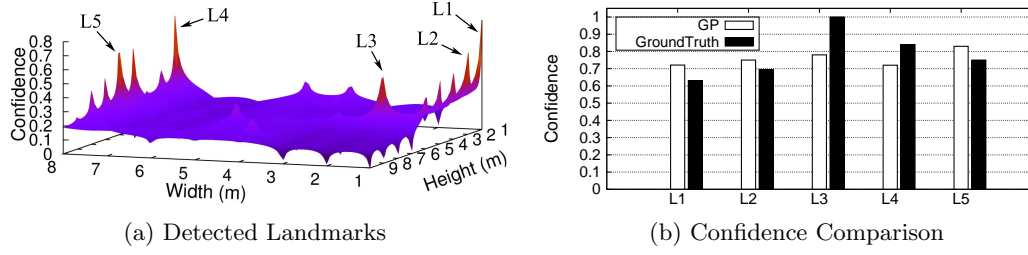


Figure 5.9: Landmark Detection

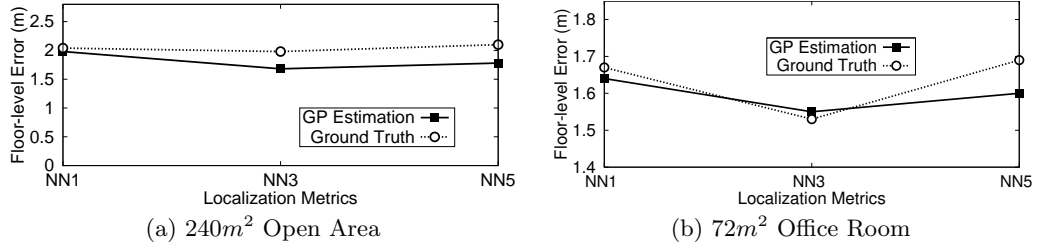


Figure 5.10: Localization Algorithm Selection

5.3.2.3 BSSID Subset Selection

Figure 5.6 and Figure 5.7 show the results of BSSID subset selection in the two different indoor environments. In total, 304 BSSIDs can be received in the open area, and 170 BSSIDs can be received in the office room. For the random selection, each BSSID is added sequentially, based on its address to the subset and the floor-level error is calculated using the selected subset. For the heuristic-based subset selection, the duplicated BSSIDs are eliminated, reducing the subset size by about half. The error rate decreases much faster than the random selection, which makes it much more efficient than the random selection algorithm if we want to achieve a floor-level error of less than three meters, the heuristic-based selection uses only 10 BSSIDs, 80% fewer than the random selection in the big open area, which requires 50 BSSIDs. In the office room, the heuristic-based selection needs only three BSSIDs, 67% percent fewer than the random selection, which needs nine BSSIDs to achieve the required accuracy.

In both environments, the number of errors decreases at a slower rate as more BSSIDs are used, indicating that the impact of individual BSSIDs on the accuracy becomes smaller as more BSSIDs are used. In addition, it might not always be better to use all the BSSIDs, as adding more BSSIDs can sometimes confuse the system. For example, in the smaller office room using fewer BSSIDs

can achieve an error of less than 1.7 meters, which is the error when all BSSIDs are used. In both Figure 5.6 and Figure 5.7, the GP-based error estimation provides a close estimate of the ground truth localization errors, and the selection algorithms can efficiently characterize the error behavior to help us understand the error using different subsets, and to help select the minimum BSSIDs required to achieve a certain accuracy.

5.3.2.4 Localization Algorithm Selection

The other application of accuracy awareness is the selection of localization algorithms. A localization algorithm determines the mapping from fingerprint to physical locations, and can significantly affect the final localization error. In the past, it is hard to compare different localization algorithms or metrics directly. With accuracy-awareness, the localization error can be easily estimated by varying different localization algorithms, and therefore the most accurate algorithm for an environment can thus be chosen accordingly. Figure 5.10 shows the error comparison of three different localization algorithms in two indoor environments. Three algorithms (NN1, NN3, NN5) are measured, which take the average of the top 1, 3 and 5 locations rated by Euclidean distance as the final inferred location respectively. In both environments, NN3 gives the best accuracy for both the GP estimation and the ground truth measurement, which suggests that NN3 should be the best choice for these two environments. Although the error reduction ($\sim 0.1\text{m}$) is not significant in this particular example when NN1 is replaced with NN3, the capability of error estimation provides us with an efficient method of algorithm selection and localization accuracy improvement.

5.4 Summary

In this work, we propose and evaluate accuracy awareness for fingerprint-based indoor localization systems. Gaussian processes learned from the radio map are used to characterize the fingerprints in an entire indoor environment. Based on the GP models built, fingerprint sampling and error estimation algorithm are used to estimate the localization errors. Concepts and applications of three granularities (point-level, region-level and floor-level) are discussed. The evaluation shows that the accuracy awareness proposed provides a close estimate of

the error behaviors of the localization systems, and useful applications such as landmark detection, localization algorithm selection and subset selection are enabled. As the accuracy awareness enables direct assessment of fingerprint-based localization systems and has many useful applications, it has the potential to be applied as a standard component in the development of future fingerprint-based localization systems.

Chapter 6

Conclusion and Future Work

As one of the most important types of context information, location connects the physical world with the cyber world, and there are many useful location-based applications [14, 68, 21, 73]. With the rise of smartphones, many users now carry smartphones daily. Locating smartphones and other mobile devices (e.g., tablets, smart watches, etc.) accurately and cost-effectively has therefore become more and more important, and has thus become the subject of rapidly increasing interest from both academics [56, 39, 93, 53] and various industries [8, 7, 4].

In this thesis, we investigate, design, and validate indoor localization systems that can provide accurate localization to mobile devices, while minimizing the start-up and maintenance costs. To address the existing challenges in system deployment, maintenance, and performance evaluation, we propose a systematic solution for both active and passive indoor localization, and the use of accuracy awareness to provide direct quality assessment of these systems. We design, implement and evaluate PiLoc and SpiLoc, which exploit participatory sensing and have a self-calibrating capability that results in lower start-up costs and adaptability to environmental changes. They provide accurate localization in terms of both active and passive indoor localization. To provide accuracy estimation to the radio maps dynamically bootstrapped from crowdsourcing, we propose A^2Loc , which takes the radio maps as input and generates accuracy estimation to provide feedback to systems such as PiLoc and SpiLoc. Together with the proposed PiLoc and SpiLoc, A^2Loc constitutes a systematic solution that advances the current state-of-the-art wireless indoor localization.

In the following sections, we provide a summary of our main research contributions and discuss possible future work.

6.1 Research Contributions

6.1.1 PiLoc: Self-calibrating Active Indoor Localization

Unlike the current state-of-the-art systems, PiLoc leverages participatory sensing to bootstrap the active localization database while requiring no prior knowledge of an indoor environment. PiLoc adopts the WiFi fingerprint-based localization scheme, its key novelty being that it merges the crowdsourcing input annotated with sensor readings and WiFi signal strengths to generate a map of the indoor environment, and construct the fingerprint database automatically. Unlike in previous systems, the self-calibrating capability makes PiLoc practical, and much easier to deploy and maintain without requiring prior knowledge of the indoor environment and dedicated site-surveys.

The evaluation shows that PiLoc is able to work in various types of indoor environments and can achieve localization accuracy comparable to that of systems that require dedicated calibration, with a localization error of 80% or less than three meters.

6.1.2 SpiLoc: Self-calibrating Passive Indoor Localization

SpiLoc does not require any collaboration from mobile devices. The key novelty of SpiLoc is that it leverages the novel RSS trace mapping technique to dynamically map the captured RSS traces to indoor pathways. The mapping automatically bootstraps the passive fingerprint database for localization. To the best of our knowledge, SpiLoc is the first participatory sensing based passive localization system that has self-calibrating capability and provides fine-grained passive localization.

The evaluation result shows that SpiLoc achieves an average localization error of 2.76m with low start-up and maintenance costs. Since SpiLoc requires no dedicated calibration and adapts to the environment, it can be easily deployed in dynamic environments for fine-grained passive localization.

6.1.3 A^2Loc : Accuracy Awareness of Fingerprint-based Wireless Indoor Localization

A^2Loc exploits a GP-based approach that uses as input the radio map collected and localization algorithm to be evaluated, and outputs the expected accuracy of the system. In addition, A^2Loc provides useful information, such as localization landmarks that can be used to further improve the localization accuracy. To the best of our knowledge, A^2Loc is the first systematic system to achieve accuracy awareness for fingerprint-based localization systems. It has the potential to be integrated into future fingerprint-based localization systems as a standard component to provide direct feedback about the accuracy level, and provide guidelines to achieve better accuracy.

6.2 Future Work

The following are some of the possible extensions of our work.

Self-calibrating in Open Areas. PiLoc leverages WiFi spectrum matching to merge WiFi-annotated walking trajectories and constructs the map of indoor walking paths. In open areas where people may not walk along distinct walkways, using signal spectrum information alone may fail to differentiate parallel walking paths that are not separated by sufficiently large distances. This is one limitation of PiLoc. Similarly, the opportunistic RSS trace mapping in SpiLoc works in environments where walking routes connecting landmarks follow indoor walking paths. In open areas, it becomes less feasible to infer the route users travel to perform RSS trace mapping purely based on the RSS measurements. Due to such challenges, to the best of our knowledge none of the existing participatory sensing based indoor localization systems have the self-calibration capability in open areas. We leave this as an open problem for future work to explore.

Enriching Self-constructed Floor Plans. By merging crowdsourcing input, PiLoc constructs indoor floor maps automatically. To improve the localization accuracy, it is useful to automatically annotate indoor floor maps with rich information such as stairs, escalators, elevators, doors, etc. Such information can

be directly extracted from smartphone sensor readings and treated as indoor landmarks to further correct localization errors and improve the final localization accuracy.

Continuous Passive Tracking. SpiLoc leverages WiFi monitors to capture transmissions from smartphones and determines the locations of the smartphones based on the self-bootstrapped passive fingerprint database. While smartphones emit wireless transmissions during WiFi communications, the transmissions can become sparse when the phones enter the sleep state. We have shown in this thesis that while the sparsity of transmission detections affects the RSS trace mapping accuracy in the bootstrapping phase, the RSS trace mapping remains robust even when the smartphone detections are sparse. The detection sparsity does not affect the instant localization as locations are determined every time the transmissions are detected. However, if the applications require the continuous tracking of smartphones, the sparsity of detection will result in poor tracking performance. Several techniques have been proposed in the literature to prompt additional phone transmissions. For example, one useful technique proposed in [56] is to let the WiFi monitors emulate popular SSIDs, and other useful techniques include sending RTS to trigger CTS responses [56]. Prompting additional transmissions from smartphones will increase both the RSS trace mapping performance and passive tracking accuracy. We leave this improvement as a possible follow-up to our work.

Extending to Multiple Fingerprints. Although PiLoc and SpiLoc are designed for active localization and passive localization respectively, both of them rely on WiFi fingerprinting. However, WiFi fingerprints are not tightly bound to our systems. Different fingerprints, such as FM radio signals [19] or even ambient noise [11] can be integrated seamlessly into our systems as additional fingerprints and used in the localization phase. Similarly, other fingerprints, such as indoor magnetic information, can be added to the system to form more discriminative fingerprints that can be used to achieve better localization accuracy.

Novel Location-based Services. The localization systems proposed in

this dissertation, PiLoc, SpiLoc and A^2Loc provide a systematic solution for accuracy-aware self-calibrating indoor localization. Many novel location-based services can be built on top of the proposed systems. For example, with passive localization, it is possible to extract the interaction patterns of the mobile devices. By looking at the mobility patterns captured by continuous passive localization, it is possible to analyze the customer flows of some particular public places. Finally, as an important service, localization privacy can also be integrated as a middleware to anonymize or randomize the MAC addresses of mobile devices during tracking. All these novel services can be added to the existing systems proposed in this thesis. We leave them as the subjects of future research.

Bibliography

- [1] Apple ios8 mac address randomization. <http://www.imore.com/closer-look-ios-8s-mac-randomization>.
- [2] Business intelligence that maximizes return on capital. <http://www.pathintelligence.com/>.
- [3] Configuring multiple bssids. http://www.cisco.com/web/techdoc/wireless/access_points/online_help/eag/123-04.JA/1100/h_ap_howto_8.html.
- [4] Easypoint. <http://www.lambda4.com/>.
- [5] Erosion and dilatation. http://docs.opencv.org/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html.
- [6] Meshlium smartphone detection. <http://www.libelium.com/products/meshlium/smartphone-detection/>.
- [7] Navisens. <http://www.navisens.com/>.
- [8] Wifi slam. <http://www.wifislam.com/>.
- [9] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller. 3d tracking via body radio reflections. In *NSDI*. Usenix, 2013.
- [10] F. Adib and D. Katabi. See through walls with wi-fi! In *SIGCOMM*. ACM, 2013.
- [11] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *MobiCom*. ACM, 2009.
- [12] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*. IEEE, 2000.
- [13] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM*. IEEE, 2000.
- [14] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa. Signals from the crowd: Uncovering social relationships through smartphone probes. In *IMC*. ACM, 2013.

- [15] R. Battiti, M. Brunato, and A. Delai. Optimal wireless access point placement for location-dependent services. 2003.
- [16] F. Ben Abdesslem, A. Phillips, and T. Henderson. Less is more: energy-efficient mobile sensing with senseless. In *Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds*. ACM, 2009.
- [17] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. *Center for Embedded Network Sensing*, 2006.
- [18] A. Campbell, S. Eisenman, N. Lane, E. Miluzzo, and R. Peterson. People-centric urban sensing. In *Proceedings of the 2nd annual international workshop on Wireless internet*, page 18. ACM, 2006.
- [19] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *MobiSys*. ACM, 2012.
- [20] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan. Indoor localization without the pain. In *MobiCom*. ACM, 2010.
- [21] J. Chon and H. Cha. Lifemap: A smartphone-based context provider for location-based services. *Pervasive Computing*, 2011.
- [22] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *UbiComp*. ACM, 2012.
- [23] R. De Maesschalck, D. Jouan-Rimbaud, and D. L. Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 2000.
- [24] M. H. DeGroot, M. J. Schervish, X. Fang, L. Lu, and D. Li. *Probability and statistics*. Addison-Wesley Reading, MA.
- [25] S. Fang, Y. Liu, W. Shen, and H. Zhu. Where are you from?: confusing location distinction using virtual multipath camouflage. In *MobiCom*. ACM, 2014.
- [26] B. Ferris, D. Fox, and N. D. Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, 2007.
- [27] B. Ferris, D. Haehnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *In Proc. of Robotics Science and Systems*. Citeseer, 2006.
- [28] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In *MobiCom*. ACM, 2004.

-
- [29] R. J. Hodrick and E. C. Prescott. Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking*, 1997.
- [30] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. Ariadne: a dynamic indoor signal map construction and localization system. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, pages 151–164. ACM, 2006.
- [31] A. Jimenez, F. Seco, C. Prieto, and J. Guevara. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In *WISP*. IEEE, 2009.
- [32] J. Jun, Y. Gu, L. Cheng, B. Lu, J. Sun, T. Zhu, and J. Niu. Social-loc: Improving indoor localization with social sensing. In *SenSys*. ACM, 2013.
- [33] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *INFOCOM*. IEEE, 2004.
- [34] M. H. Kalos and P. A. Whitlock. *Monte carlo methods*. John Wiley & Sons, 2008.
- [35] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *SenSys*. ACM, 2010.
- [36] Y. Kim, Y. Chon, and H. Cha. Smartphone-based collaborative and autonomous radio fingerprinting. *TSMC*, 2012.
- [37] E. Koukoumidis, L. Peh, and M. Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *MobiSys*. ACM, 2011.
- [38] A. Krishnakumar and P. Krishnan. On the accuracy of signal strength-based estimation techniques. In *INFOCOM*. IEEE, 2005.
- [39] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *MobiCom*. ACM, 2014.
- [40] K. Kunze, P. Lukowicz, K. Partridge, and B. Begole. Which way am i facing: Inferring horizontal device orientation from an accelerometer signal. In *ISWC*. IEEE, 2009.
- [41] N. Lane, S. Eisenman, M. Musolesi, E. Miluzzo, and A. Campbell. Urban sensing systems: opportunistic or participatory? In *HotMobile*. ACM, 2008.
- [42] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *Communications Magazine*, 2010.
- [43] N. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. Campbell, and F. Zhao. Enabling large-scale human activity inference on smartphones using community similarity networks. *Ubicomp*, 2011.

- [44] N. D. Lane, M. Mohammad, M. Lin, X. Yang, H. Lu, S. Ali, A. Doryab, E. Berke, T. Choudhury, and A. Campbell. Bewell: A smartphone application to monitor, model and promote wellbeing. In *5th international ICST conference on pervasive computing technologies for healthcare*, 2011.
- [45] B. Li, B. Harvey, and T. Gallagher. Using barometers to determine the height for indoor positioning. In *IPIN*, 2013.
- [46] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *UbiComp*. ACM, 2012.
- [47] H. Lim, L.-C. Kung, J. Hou, and H. Luo. Zero-configuration, robust indoor localization: Theory and experimentation. In *INFOCOM*. IEEE, 2006.
- [48] H. Lim, L.-C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over ieee 802.11 wireless infrastructure. *Wireless Networks*, 2010.
- [49] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of wifi based localization for smartphones. In *MobiCom*. ACM, 2012.
- [50] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *MobiSys*. ACM, 2013.
- [51] C. Luo and M. C. Chan. Socialweaver: collaborative inference of human conversation networks using smartphones. In *SenSys*, page 20. ACM, 2013.
- [52] C. Luo, H. Hong, and M. C. Chan. Piloc: a self-calibrating participatory indoor localization system. In *IPSN*. ACM/IEEE, 2014.
- [53] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim. Sail: single access point-based indoor localization. In *MobiSys*. ACM, 2014.
- [54] G. Milette and A. Stroud. *Professional Android Sensor Programming*. Wiley, 2012.
- [55] P. Mohan, V. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*, 2008.
- [56] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *SenSys*. ACM, 2012.
- [57] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: indoor location sensing using active rfid. *Wireless networks*, 2004.
- [58] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, and T. Hart. Coin-gps: indoor localization from direct gps receiving. In *MobiSys*. ACM, 2014.
- [59] J. Niu, B. Lu, L. Cheng, Y. Gu, and L. Shu. Ziloc: Energy efficient wifi fingerprint-based localization with low-power radio. In *WCNC*. IEEE, 2013.

-
- [60] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *MobiCom*. ACM, 2007.
 - [61] N. Patwari and J. Wilson. Rf sensor networks for device-free localization: Measurements, models, and algorithms. *Proceedings of the IEEE*, 2010.
 - [62] N. B. Priyantha. *The cricket indoor location system*. PhD thesis, MIT, 2005.
 - [63] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *MobiCom*. ACM, 2012.
 - [64] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu. Ear-phone: an end-to-end participatory urban noise mapping system. In *IPSN*. ACM, 2010.
 - [65] T. S. Rappaport et al. *Wireless communications: principles and practice*. prentice hall PTR New Jersey.
 - [66] C. E. Rasmussen. Gaussian processes for machine learning. 2006.
 - [67] N. Roy, H. Wang, and R. Roy Choudhury. I am a smartphone and i can tell my user’s walking direction. In *MobiSys*. ACM, 2014.
 - [68] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjaergaard. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *PerCom*. IEEE, 2014.
 - [69] K. Sankaran, M. Zhu, X. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh. Using mobile phone barometer for low-power transportation context detection. In *SenSys*, 2014.
 - [70] M. Seifeldin, A. Saeed, A. Kosba, A. El-Keyi, and M. Youssef. Nuzzer: A large-scale device-free passive localization system for wireless environments. *TMC*, 2013.
 - [71] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef. Nuzzer: A large-scale device-free passive localization system for wireless environments. *TMC*, 2013.
 - [72] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: spot localization using phy layer information. In *MobiSys*. ACM, 2012.
 - [73] A. Serra, D. Carboni, and V. Marotto. Indoor pedestrian navigation system using a modern smartphone. In *MobiHCI*. ACM, 2010.
 - [74] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang. Walkie-markie: indoor pathway mapping made easy. In *NSDI*. USENIX, 2013.
 - [75] W. Sun, J. Liu, C. Wu, Z. Yang, X. Zhang, and Y. Liu. Moloc: On distinguishing fingerprint twins. In *ICDCS*. IEEE, 2013.

- [76] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys*. ACM, 2012.
- [77] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *TOIS*, 1992.
- [78] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *TMC*, 2010.
- [79] J. Wilson and N. Patwari. See-through walls: Motion tracking using variance-based radio tomography networks. *TMC*, 2011.
- [80] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *IPSN*. IEEE, 2014.
- [81] J. Xiong and K. Jamieson. Arraytrack: a fine-grained indoor location system. *HotMobile*, 2012.
- [82] C. Xu, B. Firner, R. S. Moore, Y. Zhang, W. Trappe, R. Howard, F. Zhang, and N. An. Scpl: indoor device-free multi-subject counting and localization using radio signal strength. In *IPSN*. ACM, 2013.
- [83] C. Xu, B. Firner, Y. Zhang, R. Howard, J. Li, and X. Lin. Improving rf-based device-free passive localization in cluttered indoor environments through probabilistic classification methods. In *IPSN*. ACM/IEEE, 2012.
- [84] N. Xu, K. H. Low, J. Chen, K. K. Lim, and E. B. Ozgul. Gp-localize: Persistent mobile robot localization using online sparse gaussian process observation model. *AAAI*, 2014.
- [85] Y. Yang and A. E. Fathy. See-through-wall imaging using ultra wideband short-pulse radar system. In *Antennas and Propagation Society International Symposium*. IEEE, 2005.
- [86] Z. Yang, C. Wu, and Y. Liu. Locating in fingerprint space: wireless indoor localization with little human intervention. In *MobiCom*. ACM, 2012.
- [87] J. Y. Yen. Finding the k shortest loopless paths in a network. *management Science*, 1971.
- [88] M. Youssef and A. Agrawala. The horus wlan location determination system. In *MobiSys*. ACM, 2005.
- [89] M. Youssef and A. Agrawala. The horus wlan location determination system. In *MobiSys*. ACM, 2005.
- [90] M. Youssef, M. Mah, and A. Agrawala. Challenges: device-free passive localization for wireless environments. In *MobiCom*. ACM, 2007.

- [91] M. A. Youssef and A. Agrawala. On the optimality of wlan location determination systems. 2003.
- [92] D. Zhang, J. Ma, Q. Chen, and L. M. Ni. An rf-based system for tracking transceiver-free objects. In *PerCom*. IEEE, 2007.
- [93] Y. Zheng, G. Shen, L. Li, C. Zhao, M. Li, and F. Zhao. Travi-navi: Self-deployable indoor navigation system. In *MobiCom*. ACM, 2014.